

OPTIMAL POWER FLOW VIA QUADRATIC MODELING

A Dissertation
Presented to
The Academic Faculty

by

Ye Tao

In Partial Fulfillment
Of the Requirements for the Degree
Doctor of Philosophy in the
School of Electrical and Computer Engineering

Georgia Institute of Technology
December 2011

Copyright © Ye Tao 2011

OPTIMAL POWER FLOW VIA QUADRATIC MODELING

Approved by:

Dr. A. P. Sakis Meliopoulos, Advisor
School of Electrical and Computer
Engineering
Georgia Institute of Technology

Dr. Carlos S. Grijalva
School of ECE
School of Electrical and Computer
Engineering
Georgia Institute of Technology

Dr. Bonnie H. Ferri
School of Electrical and Computer
Engineering
Georgia Institute of Technology

Dr. Shijie Deng
School of Industrial and Systems
Engineering
Georgia Institute of Technology

Dr. Yorai Y. Wardi
School of Electrical and Computer
Engineering
Georgia Institute of Technology

Date Approved: Aug 22, 2011

ACKNOWLEDGEMENTS

The doctoral study at Georgia Tech is a long-distance adventure. This journey would not have been accomplished without the inspiration, encouragement, and help from many people. I would like to express my heartfelt gratitude to all of them at this time.

First of all, I would like to extend my most sincere thanks to Dr. A. P. Sakis Meliopoulos. I could not have imagined having a better advisor and mentor during tough times in the Ph.D. pursuit. It is due to his patience, motivation, inspiration, and immense knowledge that I have gained a deeper understanding of engineering and mathematics. His careful guidance on solving problems, writing reports and oral communicating with profession rewards my entire research life. Without his illuminating instruction, consistent encouragement, and confidence in my capability, this thesis work could not be completed in the present form. I am deeply grateful for his guidance and support during my Ph. D. life.

I am grateful to Dr. A. P. Sakis Meliopoulos and Dr. Carlos S. Grijalva for their knowledge and experience in my interactions with them, and especially for their suggestions and guidance on my research work. I would like to thank Dr. Bonnie Heck Ferri, Dr. J. Yorai Wardi, and Dr. Shijie Deng for being my dissertation committee members and for their support in completion of this work.

I owe many thanks to Frank Lambert at NEETRAC for his invaluable contribution to my research, which would not have been completed without his constant help and support. I would like to specially thank David Qu, Gail Palmer, and Dr. Karen Head for their guidance and help on my writing and communication skills. I am also heartily thankful to Dr. George Cokkinides for his help and assistance to my experimental work. My gratitude are also given to Dr. George K. Stefopoulos and Dr. Q. Binh Dam,

who gave me valuable information about the electrical power system analysis methods and data used in industry.

I would like to thank the editors Mitch Costley, Xin Du, Vangelis Farantatos, Dr. Stefan Grubic, Dustin Howard, Andrew Paquette, Liangyi Sun, and Aniemi Umana for their support on grammar and writing style.

The members of the Power Systems Control and Automation Laboratory (PSCAL) have contributed generously to my personal and research time at Georgia Tech. The people in the group are friendly and always offer me good advice and collaboration. I would like to thank Fan Cai, Yongnam Cho, Vangelis Farantatos, Stephanie M. Gossman, Renke Huang, Kyle Howells, Anupama Keeli, Yonghee Lee, Meijun Liu, Sandhya Madan, Stephane Ntwoku, Jongkook Park, Curtis Roe, Liangyi Sun, Aniemi Umana, and Xuebei Yu for their friendship and support. A special thanks to Renke Huang, who generously shared with me his valuable experience with WinIGS software and power system analysis experience.

Furthermore, I am also pleased to extend my thanks to all graduate students in electrical energy group. A special thank goes to Dr. Yao Duan and his wife Qin Sun. I would also like to bring my appreciation to Siwei Cheng, Mitch Costley, Jing Dai, Liang Du, Dawei He, Dustin Howard, Dr. Guangfei Geng, Dr. Stefan Grubic, Shalini Gupta, Jiaqi Liang, Dr. Salman Mohagheghi, Andrew Paquette, Anish Prasai, Zhengkai Wu, Dr. Yi Yang, and Dr. Pinjia Zhang.

I am thankful to all my friends at Georgia Tech. They supported me academically, personally, and especially during difficulties. Among them special thanks go to Dr. Tianci Jiang and her brother Dr. Tianyue Jiang, Shicong Meng and his wife Wei Zhuo, Min Yang, Xin Zhao and her husband Hongyi Qu. I also would like to mention Dr. Jiang Bian, Xin Chen, Jie Dang, Dr. Dennis Ding, Chengyu Guan, Yu Gao, Shuang Hao, Pingping He, Dr. Michael Healy, Neale Hightower and his wife Carol Hightower, Dan Hou and his wife Na Lei, Nan Hua, Dr. Gang Huang, Lingbin Kong, Dr. Fang Li, Qing

Li, Dr. Qinzhen Liang, Qifeng Lin, Xiao Liu, Dr. Xuyuan Liu, Yu Liu and his wife Vivian Yu, Dr. Bo Lu, Long Lu, Dr. Nan Lu, Chengchi Luo, Dexin Luo, Mengdi Luo, Dae Hyun Kim, Siyu Ji, Dr. Jun Ma, Xiaoyao Ma, Jennifer Pan, Siyang Peng, Cong Shi, Xusheng Sun, Ge Song, Jie Tan and his wife Yuting Gu, Chao Wang and his wife Zhenxian Wang, Qingyang Wang, Dr. Ting Wang, Xiaojia Wang, Xing Wang, Eric Wong, Tan Wu, Dr. Chunpeng Xiao, Jiao Xu, Linji Yang and his wife Pengyi Shi, Dr. Ping Yang, Ying Yang, Hong Yu, Xuehong Yu, Dongbo Zhang, Junjie Zhang, Min Zhang, Dr. Rongwei Zhang, Yaan Zhang, Yimin Zhang, Yu Zhang, Gaoyuan Zhao, George Jin Zhao, Xiaonan Zhao, Dr. Pan Zhou, Ran Zhou, and Guomei Zhu.

There are numerous names of faculty, family and friends that I should mention who have helped me during my five years in Atlanta, Georgia. In particular I would like to sincerely appreciate Xin Du and Liyue Fan. I also want to express my gratitude to Xianyu Chen, Zhang Huang, Haoran Li, Qiaoling Liu, Shan Luo, Mingxuan Xie, Qiaoqiao Xu, Xue Zhou, Anda Wang, Haowei Wang, and Ling Weng.

Most of all, I owe my loving thanks to my family. My parents' encouragement and support kept me focused and motivated. With their unflagging love, encouragement, and understanding, I eventually reached the final stages of this doctoral degree.

The generous financial supports from the following institutions/organizations are greatly appreciated:

- National Electric Energy Testing Research and Applications Center (NEETRAC)
- US Department of Energy
- Power System Engineering Research Center (PSERC)

TABLE OF CONTENTS

	Page
ACKNOWLEDGEMENTS	iii
LIST OF TABLES	xii
LIST OF FIGURES	xiv
SUMMARY	xvii
CHAPTER 1	
INTRODUCTION AND OBJECTIVES OF RESEARCH	1
1.1 Problem Statement	1
1.2 Research Objectives	2
1.3 Thesis Outline	4
CHAPTER 2	
LITERATURE REVIEW AND BACKGROUND INFORMATION	6
2.1 Introduction	6
2.2 The Development of the OPF Problem	6
2.3 The Algorithm Classification of the OPF Problem	8
2.3.1 Nonlinear Programming	8
2.3.2 Intelligent Search Algorithms	11
2.3.3 Sequential Algorithms	12
2.3.4 Decomposition Technologies	14
2.4 Power System Modeling	15
2.5 Three-Phase Optimal Power Flow	16
2.6 Linearization Techniques	17
2.7 Summary	18
CHAPTER 3	
QUADRATIC POWER SYSTEM MODELING	20
3.1 Quadratic Power System Modeling	20

3.1.1	Overview	20
3.1.2	Quadratic General Bus Modeling	21
3.2	Summary	28
CHAPTER 4		
OPTIMAL POWER FLOW PROBLEM DEFINITION		29
4.1	Introduction	29
4.2	The Quadratic Problem Formulation	29
4.3	Summary	32
CHAPTER 5		
THE PROPOSED OPTIMAL POWER FLOW ALGORITHM		33
5.1	Introduction	33
5.2	Algorithm Outline	33
5.3	SLP Algorithm Implementation	35
5.3.1	Initialization	35
5.3.2	Define the Optimization Problem	39
5.3.3	Form the Linearized Optimization Problem	39
5.3.4	Solve the System	42
5.3.5	Eliminate Violations in the Modeled System	43
5.3.6	Procedures for Solving the Violation for All Constraints	45
5.3.7	Procedures for the Next Iteration	45
5.4	SQP Algorithm Implementation	46
5.5	Parallelism in OPF	47
5.6	Post-Solution Sensitivity Analysis	49
5.7	The OPF Software Design	50
5.8	Summary	52
CHAPTER 6		

THREE-PHASE OPTIMAL POWER FLOW	54
6.1 Introduction	54
6.2 Three-Phase OPF	54
6.3 Three-Phase Quadratic General Bus Modeling	56
6.3.1 The Description of OPF State and Control Variables	58
6.3.2 The Description of OPF Mismatch Variables	62
6.4 TOPF Constraint Description	62
6.5 Algorithm Description	64
6.5.1 Assign Initial Values of x and u to x^o and u^o	64
6.5.2 Solve the Linearized Optimization Problem $v, \Delta u, \Delta I_m$	68
6.5.3 The Branch and Bound Algorithm to TOPF	68
6.6 The TOPF Software Design	70
6.7 Summary	72
CHAPTER 7	
A TOPF APPLICATION – OPTIMAL VAR ALLOCATION WITH DYNAMIC PROGRAMMING	73
7.1 Introduction	73
7.2 Cost Definitions	74
7.2.1 The Annualized Equivalent Cost	75
7.2.2 The Operating Cost	76
7.2.3 The Voltage Deviation Penalty	77
7.2.4 The Voltage Recovery Time Penalty	77
7.2.5 The Voltage Oscillation Penalty	78
7.2.6 Hard Constraint 1 – The Voltage Lower Bound	79
7.2.7 Hard Constraint 2 – The Voltage Recovery Time Upper Bound	79
7.3 Candidate Location Selection	79

7.4	The Formation of States	81
7.5	The Optimization Problem Definition	81
7.6	The Algorithm Structure	89
7.7	Summary	90
CHAPTER 8 DEMONSTRATION AND EVALUATION OF PROPOSED OPF METHOD WITH SEVERAL TEST SYSTEMS		91
8.1	Introduction	91
8.2	A Three-Bus System Example	91
8.2.1	The Solution with Polar Power Flow	93
8.2.2	The Solution with Quadratized Power Flow	96
8.3	The RTS-79 System Example	101
8.4	The RTS-96 System Example	107
8.5	Test Systems with Different Sizes	113
8.6	Post-Solution Sensitivity Analysis	117
8.7	Summary	119
CHAPTER 9 DEMONSTRATION AND EVALUATION OF PROPOSED TOPF WITH SEVERAL TEST SYSTEMS		120
9.1	Introduction	120
9.2	An Three-Phase Eight-Bus System Example	120
9.3	The Three-Phase RTS-79 System Example	123
9.4	The Three-Phase RTS-96 System Example	125
9.5	Summary	127
CHAPTER 10 DEMONSTRATION OF REACTIVE SOURCE PLANNING WITH DYNAMIC PROGRAMMING AND THE PROPOSED OPF METHOD		128
10.1	Introduction	128

10.2	System Description	128
10.3	Candidate Reactive Sources	131
10.4	Candidate Locations Selection	131
10.5	State Definitions	132
10.6	Simulation Results	133
10.7	Cost Evaluation	135
10.7.1	Transition costs between Stages	135
10.7.2	Operating Costs computed via TOPF	136
10.7.3	Costs at Stage 19	138
10.7.4	Examples of Cost Evaluations	138
10.8	The Computational Example of Dynamic Programming	141
10.9	The Planning Details of Optimal VAR Allocation	142
10.10	Summary	146
CHAPTER 11		
CONCLUSIONS AND FUTURE RESEARCH DIRECTION		148
11.1	Conclusions	148
11.2	Future Work	149
APPENDIX A		
QUADRATIC SINGLE-PHASE TRANSFORMER MODEL		152
APPENDIX B		
LINEARIZATION METHODS		154
B.1	Overview	154
B.2	The Definition of the Derivative	155
B.3	The Co-state Method	155
B.4	Linearization Update Methods	156
B.5	Linearization Limit Strategies	159
REFERENCES		162

VITA	175
------------	-----

LIST OF TABLES

	Page
Table 8-1: Unit parameters in actual units in the three-bus system	91
Table 8-2: Unit parameters in per-unit scale in the three-bus system	91
Table 8-3: The operating constraints added at each iteration for the RTS-79 system	105
Table 8-4: Runtime information of the RTS-79 system without parallelism	106
Table 8-5: Runtime information of the RTS-79 system with parallelism	107
Table 8-6: The operating constraints added at each iteration for the RTS-96 system	110
Table 8-7: Runtime information of the RTS-96 system without parallelism	111
Table 8-8: Runtime information of the RTS-96 system with parallelism	112
Table 8-9: Runtime information of nine cases	113
Table 9-1: The load parameters in actual units in the eight-bus system	122
Table 9-2: Runtime information of the eight-bus system	123
Table 9-3: Runtime information of the three-phase RTS-79 system	125
Table 9-4: Runtime information of the three-phase RTS-79 system	127
Table 10-1: Candidate reactive sources	131
Table 10-2: Locations of capacitor bank with sensitivity calculation	132
Table 10-3: State definitions	133
Table 10-4: Simulation data of Substation 1 for State 6 at Stage 19	134
Table 10-5: Simulation data of Substation 2 for State 6 at Stage 19	135
Table 10-6: Transition Costs, Stage $k-1$ to Stage k	136
Table 10-7: TOPF results from Stage 0 to Stage 20	137
Table 10-8: Cost data for all states at Stage 19	138
Table 10-9: Cost data for State 6 at Stage 19	141
Table 10-10: Optimal trajectory costs from Stage 0 to all states at Stage 18	141

LIST OF FIGURES

	Page
Figure 3.1 A general power system bus	21
Figure 5.1 The flow chart of the SLP algorithm	34
Figure 5.2 A generic quad-core processor	48
Figure 5.3 OPF flow chart update for parallelism	49
Figure 5.4 The OPF software design	51
Figure 6.1 The one-line diagram of a general bus in three-phase power systems	57
Figure 6.2 The TOPF software design	70
Figure 7.1 Cost classification	75
Figure 7.2 Decision tree: at each stage, there are 16 states	82
Figure 7.3 Transitions from Stage $k-1$ to Stage k	86
Figure 7.4 Decision tree: at each stage, there are n states and m stages	87
Figure 7.5 The reactive planning algorithm	89
Figure 8.1 A three-bus power system	92
Figure 8.2 The cost with mismatches for the three-bus system using the polar power flow	95
Figure 8.3 The cost without mismatches for the three-bus system using the polar power flow	95
Figure 8.4 The variables of the three-bus system using the polar power flow	96
Figure 8.5 The cost with mismatches for the three-bus system using the quadratized power flow	99
Figure 8.6 The cost without mismatches for the three-bus system using the quadratized power flow	99
Figure 8.7 Variables for the three-bus system using the quadratized power flow	100
Figure 8.8 The IEEE RTS-79 power system	101
Figure 8.9 The cost with mismatches for the RTS-79 system	103

Figure 8.10 The cost without mismatches for the RTS-79 system	103
Figure 8.11 The real power loss of the RTS-79 system	104
Figure 8.12 The cost with mismatches for the RTS-96 system	108
Figure 8.13 The cost without mismatches for the RTS-96 system	109
Figure 8.14 The real power loss of the RTS-96 system	109
Figure 8.15 The costs with/without mismatches for the 6-bus system	114
Figure 8.16 The costs with/without mismatches for the 9-bus system	115
Figure 8.17 The costs with/without mismatches for the 14-bus system	115
Figure 8.18 The costs with/without mismatches for the 24-bus system	115
Figure 8.19 The costs with/without mismatches for the 30-bus system	116
Figure 8.20 The costs with/without mismatches for the 39-bus system	116
Figure 8.21 The costs with/without mismatches for the 57-bus system	116
Figure 8.22 The costs with/without mismatches for the 118-bus system	117
Figure 8.23 The costs with/without mismatches for the 300-bus system	117
Figure 8.24 Sensitivity analysis on the upper bound of a transformer-tap variable in the RTS-96 system	118
Figure 8.25 Sensitivity analysis on the lower bound of a real-power variable for the RTS- 96 system	119
Figure 9.1 An eight-bus power system	120
Figure 9.2 The transmission line (BUS1H to BUS2H) parameters in the eight-bus power system	121
Figure 9.3 The costs with/without mismatches for the eight-bus system	122
Figure 9.4 The costs with/without mismatches for the three-phase RTS-79 system	123
Figure 9.5 The three-phase RTS-79 power system	124
Figure 9.6 The sketch map of the three-phase RTS-96 power system	126
Figure 9.7 The costs with/without mismatches for the three-phase RTS-96 system	126

Figure 10.1 The test system for dynamic VAR planning	129
Figure 10.2 Substation 1 in the test system for dynamic VAR planning	130
Figure 10.3 Substation 2 in the test system for dynamic VAR planning	130
Figure 10.4 The Phase A voltage of Bus S1-MCC_13 at State 3, Stage 2	140
Figure 10.5 The optimal transitions from Stage 0 to Stage 6	143
Figure 10.6 The optimal transitions from Stage 7 to Stage 13	144
Figure 10.7 The optimal transitions from Stage 14 to Stage 20	145
Figure A.1 The single-phase transformer model	152
Figure B.1 The result reaches the boundary of a constraint	157
Figure B.2 The result does not reach the boundary of the constraint	158
Figure B.3 The final version of the linearization update method	159

SUMMARY

Optimal power flow (OPF) is the choice tool for determining the optimal operating status of the power system by managing controllable devices. The importance of the OPF approach has increased due to increasing energy prices and availability of more control devices. Existing OPF approaches exhibit shortcomings. Current OPF algorithms can be classified into (a) nonlinear programming, (b) intelligent search methods, and (c) sequential algorithms. Nonlinear programming algorithms focus on the solution of the Kuhn-Tucker conditions; they require a starting feasible solution and the model includes all constraints; these characteristics limit the robustness and efficiency of these methods. Intelligent search methods are first-order methods and are totally inefficient for large-scale systems. Traditional sequential algorithms require a starting feasible solution, a requirement that limits their robustness. Present implementations of sequential algorithms use traditional modeling that result in inefficient algorithms.

The research described in this thesis has overcome the shortcomings by developing a robust and highly efficient algorithm. Robustness is defined as the ability to provide a solution for any system; the proposed approach achieves robustness by operating on suboptimal points and moving toward feasible, it stops at a suboptimal solution if an optimum does not exist. Efficiency is achieved by (a) converting the nonlinear OPF problem to a quadratic problem (b) and limiting the size of the model; the quadratic model enables fast convergence and the algorithm that identifies the active constraints, limits the size of the model by including only the active constraints.

A concise description of the method is as follows: The proposed method starts from an arbitrary state which may be infeasible; model equations and system constraints are satisfied by introducing artificial mismatch variables at each bus. Mathematically this is an optimal but infeasible point. At each iteration, the artificial mismatches are reduced while the solution point maintains optimality. When mismatches reach zero, the solution

becomes feasible and the optimum has been found; otherwise, mismatch residuals are converted to load shedding and the algorithm provides a suboptimal but feasible solution. Therefore, the algorithm operates on infeasible but optimal points and moves towards feasibility.

The proposed algorithm maximizes efficiency with two innovations: (a) quadratization that converts the nonlinear model to quadratic with excellent convergence properties and (b) minimization of model size by identifying active constraints, which are the only constraints included in the model. Finally sparsity technique is utilized that provide the best computational efficiency for large systems.

This dissertation work demonstrates the proposed OPF algorithm using various systems up to three hundred buses and compares it with several well-known OPF software packages. The results show that the proposed algorithm converges fast and its runtime is competitive.

Furthermore, the proposed method is extended to a three-phase OPF (TOPF) algorithm for unbalanced networks using the quadratized three-phase power system model. An example application of TOPF is presented. Specifically, TOPF is utilized to address the problem of fault induced delayed voltage recovery (FIDVR) phenomena, which lead to unwanted relay operations, stalling of motors and load disruptions. This thesis presents a methodology that will optimally enhance the distribution system to mitigate/eliminate the onset of FIDVR. The time-domain simulation method has been integrated with a TOPF model and a dynamic programming optimization algorithm to provide the optimal reinforcing strategy for circuits.

CHAPTER 1

INTRODUCTION AND OBJECTIVES OF RESEARCH

1.1 Problem Statement

Optimal power flow (OPF) computes the optimal operating status of a power system with respect to the controllable devices. Since it is highly efficient and accurate, OPF is widely used in power system operation and planning. Efficient OPF software needs to solve both the operation problem and the planning problem. Operational OPF usually runs in energy management systems and is used to solve the optimization problem in a time duration of minutes, hours, or up to one day. Therefore, operational OPF requires high convergence speed. Planning OPF is a planning tool that is used to maximize the capability of the existing system assets [5] for a planning period usually of five to twenty years. This planning problem can be separated in stages (e.g., one year) and OPF is used to compute the operating costs in each stage, so OPF is a subproblem in this planning problem.

OPF is formed as an optimization process to minimize or maximize a certain objective function of the power system while satisfying system constraints. The objectives usually include the minimum thermal unit cost [1], the minimum transmission loss [2], the maximum system loadability [3], the maximum reactive reserve margin [4], and so on. The system constraints limit transmission flows, bus voltage magnitudes, the real and reactive powers of generators, and some other physical quantities of the system. All these objective functions and constraints can be functionally represented by the control variables and the state variables of the system.

1.2 Research Objectives

The OPF problem has been studied for more than 40 years. The importance of the OPF approach has increased due to increasing energy prices and demand. Since Carpentier [10] defined the OPF problem in 1962, many algorithms have been designed to solve the base OPF problem and its derivative problems [1]-[4], [6]-[9], [11]-[13]. Current OPF algorithms can be classified into the following categories: sequential algorithms [2], nonlinear programming algorithms [11], and intelligent search methods [29]-[31]. These algorithms still have some shortcomings. Traditional sequential algorithms cannot solve for an infeasible system since they need a feasible operating point as the initial solution; in addition, they use the traditional power system model that results in inefficient algorithms. Nonlinear programming algorithms require a starting feasible solution, and the solving model includes all constraints. Therefore, the robustness and efficiency of these methods are limited. Intelligent search methods have bloomed in recent years; they are first-order methods and are totally inefficient for large-scale power systems.

The objective of this thesis is to develop a robust and highly efficient algorithm for OPF. Robustness is achieved by the capability of handling both feasible and infeasible systems. We propose a method that starts from an arbitrary state that may be infeasible by introducing artificial mismatch variables at each bus to eliminate the violations in model equations and system constraints. This initial operating point is optimal since the algorithm can select specific initial values of the variables to minimize the objective function. The algorithm reduces artificial mismatches iteratively while maintaining the optimal solution point. The optimization problem in each iteration is converted to a linear programming (LP) problem using the co-state method. To mitigate the linearization errors, a linearization limit constraint is added for each control variable in the LP problem; otherwise, the LP solution may not induce a valid power flow solution. In addition, some operating constraints in the power flow solution are out of bounds even when they are

already included. To solve this issue and ensure artificial feasibility, the algorithm updates the \mathbf{b} vector in the LP problem according to overshoots, retrieves the previous solution, and solves the updated LP problem. When the mismatches are reduced to zero, the solution becomes feasible and the optimum has been found; otherwise, the mismatch residuals are converted into remedial measures (example: load shedding), and the algorithm provides a suboptimal but feasible solution. Sometimes, one or two more iterations are needed after mismatches are reduced to zero due to linearization errors in the final iteration. Therefore, the algorithm operates on infeasible but optimal points and moves towards feasibility. Efficiency is achieved by properly reducing the problem size. The algorithm maximizes efficiency with three methods: (a) converts the nonlinear power system model to quadratic for excellent convergence properties, (b) identifies active constraints and adds only those to the model, and (c) uses sparsity techniques to provide the best computational efficiency for large-scale systems. This OPF algorithm has already been applied on a three-bus system, several IEEE test systems including the RTS-79 system [76], the RTS-96 system [77], and several other well-known benchmark systems of sizes from six buses to three-hundred buses. The three-bus test example demonstrates the algorithm flow using polar power flow and quadratized power flow.

Another important task of the proposed work is extending this OPF algorithm to three-phase power systems using a three-phase quadratic model. Since smart grid technologies are blooming in recent years, more and more research works is focusing on unbalanced distribution networks. However, traditional OPF tools do not fit for distribution networks since they are designed for balanced transmission networks. This work proposed a three-phase OPF (TOPF) algorithm modified from the proposed OPF algorithm. TOPF keeps the main flow unchanged since the algorithm structure and device structures in the proposed OPF algorithm are independent. However, the software design and implementation of TOPF are different from single-phase OPF since the three-phase power system model is much more complicated than the symmetric and balanced power

system model. TOPF is demonstrated using an eight-bus system, the RTS-79 system, the RTS-96 system, and a system for optimal VAR source placement.

The third task is applying the proposed TOPF algorithm to an optimal VAR source planning problem on a power system with both distribution and transmission networks. The objective of the planning problem is to mitigate or eliminate fault induced delayed voltage recovery (FIDVR) phenomena by strategically placing static and dynamic VAR resources. The costs in the planning problem include the investment cost, the installation cost, the operating cost, the voltage deviation penalty, the recovery time penalty, the oscillation penalty, and two hard-constraint penalties. The operating cost is computed using TOPF. The planning problem is solved via dynamic programming to find the VAR allocation at each stage with the minimum optimal trajectory cost from the initial stage to the final stage in the planning horizon while satisfying the performance criteria.

1.3 Thesis Outline

The outline of the remaining document is as follows:

Chapter 2 introduces the history of the OPF problem and presents the origin and research branches. The origin and description of the OPF problem are presented at the beginning of Chapter 2, followed by a substantial literature survey organized by the different problem setups and algorithms. Chapter 2 also gives some reviews on power system modeling, three-phase OPF algorithms, and linearization techniques.

Chapter 3 describes the quadratic modeling of symmetric and balanced power systems represented with per phase equivalents. This model uses quadratized power flow with Kirchhoff's current law (KCL) and the Cartesian coordinate system. Quadratization can provide better convergence properties in the proposed algorithm.

Chapter 4 elaborates on the quadratized OPF problem formulation with only linear and quadratic items. This formulation introduces mismatch variables on every bus

in the power system, so the proposed algorithm can start from an arbitrary working point whether feasible or infeasible.

Chapter 5 shows the design and implementation of the proposed OPF algorithm using sequential methods including sequential linear programming (SLP) and sequential quadratic programming (SQP). Furthermore, this chapter also describes several related topics, such as parallelism, post-solution sensitivity analysis, and the software design.

Chapter 6 is a description of TOPF topics including problem definition, modeling, algorithm description, and the software design. Since the TOPF algorithm is modified from the proposed OPF algorithm, this chapter presents these modifications only.

Chapter 7 gives an application to the TOPF algorithm: the optimal VAR planning problem solved via the dynamic programming method. TOPF computes the operating costs in the planning process.

Chapter 8-10 presents examples, solutions, and analysis for OPF, TOPF and the optimal VAR planning problem respectively.

Finally, Chapter 11 provides the summary of the thesis work and some future research orientations.

This dissertation has two appendices. Appendix A presents the detailed information of a single-phase transformer model. Appendix B presents works related to linearization.

CHAPTER 2

LITERATURE REVIEW AND BACKGROUND INFORMATION

2.1 Introduction

OPF includes a class of optimization problems pursuing a specific objective while satisfying operating and physical constraints to maintain electric power system operation. Since the first OPF has been proposed [10], it has become a crucial topic in power system operation, and many derived problems and algorithms have also been developed. As computer and computation technologies develop and energy savings become a significant issue in the modern and future world, OPF formulation becomes more and more complicated, large scale, and realistic. This chapter summarizes up-to-date OPF formulations and algorithms.

2.2 The Development of the OPF Problem

As power systems become more complicated and economically sensitive, OPF also becomes more complex, realistic, and efficient. These developments are summarized in the following paragraphs:

Several decades ago, researchers modeled a power system by DC power flow [15] for fast computing. Then, they solved OPF using AC power flow, a more accurate and complicated model, thanks to developments of computer and computational technologies.

Power systems are traditionally modeled in polar coordinates [1], but the rectangular model has become more and more important in recent years [17] [18] because of its fast convergence speed when solving the power flow. However, it contains more power flow equations.

The objective function also has many diverse realizations. At the beginning, researchers focused on minimizing the loss only [14]. However, the minimum loss does

not mean the minimum cost, since different fuels have different rates and efficiencies [11]. Nowadays, minimizing the fuel cost is the most popular objective. For different purposes, the objective of OPF can also be the loadability of the system, the voltage margins at load buses, the reactive power reserve margins [4], etc. In addition, multiple objectives are used to balance different goals [6]. Different combined objectives show the different importance to each item. For example, incorporating voltage stability or desiring larger load margin may result in higher operating cost.

The variables in OPF were all continuous in the early years, since continuous problems can be solved by high efficient optimization algorithms such as linear programming and Newton's methods. Contrarily, the discrete optimization problem is NP-hard and usually is solved by exponential-time algorithms, e.g., dynamic programming [16]. One method to improve the computing speed is to use some advanced algorithms in the mixed-integer programming category. For example, Gomez et al. introduced a new discrete VAR source model to the OPF problem and solved the planning algorithm using the decomposition method and the branch and bound algorithm [17] in the early 1990s. Although the mixed-integer programming problem is also NP-hard, these methods can reduce runtime for some special defined problems.

A power system has many restrictions on operating states for security purposes, such as bus voltage magnitudes, generator active/reactive powers, controllable transformer ratios, etc. However, these restrictions cannot ensure safe running when contingencies occur. Therefore, OPF is extended to security-constrained optimal power flow (SCOPF) to ensure that the power system runs at its safe region when a contingency occurs [19], [20]. SCOPF can be classified into preventive mode, corrective mode, and preventive/corrective mode. In preventive mode, the solution is secure in both the base case and post-contingency cases. Therefore, preventive SCOPF includes constraints in both base case OPF and post-contingency OPF, so its constraint set is much larger than the constraint set of OPF. In corrective mode, the solution is permitted to adjust after any

contingency occurs. Therefore, corrective SCOPF includes fewer constraints than preventive SCOPF does but is less secure [7]. The number of constraints in preventive/corrective mode is intermediate between the previous two modes. Several decomposition methods are proposed to solve SCOPF, and post-contingency OPF is usually viewed as a slave problem of the base case OPF [21].

The stability of the power system is usually ensured by operating constraints, such as voltage constraints. However, the operating constraints are not a sufficient and necessary condition of system stability. There are always some exceptions. Therefore, some researchers incorporate stability constraints described by the generator transient model into OPF directly [22]. Usually the transient model of generators is described by differential equations and transformed into algebra equations via numerical methods, such as the Runge-Kutta method [8].

2.3 The Algorithm Classification of the OPF Problem

Since OPF is nonlinear, nonconvex, large-scale, and possibly discrete, nearly all optimization methods have been tried. For example, the interior-point method (IPM) developed in the 1950s [24] and 1960s [25] has become a very important method in solving OPF since the 1990s [3], [26]-[28]. In addition, once intelligent search algorithms, such as the genetic algorithm [29], the particle swarm algorithm [30], and artificial neural network [31], were successfully used in other optimization problems, researchers quickly introduced those to OPF. This section outlines some significant works in OPF according to algorithm types.

2.3.1 Nonlinear Programming

Nonlinear optimization problems are usually transformed into unconstrained problems (equations of the Kuhn-Tucker conditions) by a Lagrangian function, the Powell method [32], [33], Sequential Unconstrained Minimization Technique (SUMT)

[34], the MINOS augmented concept [35], IPM [57], etc. Then, the transformed problem can be solved by the gradient method [11], the Newton-Raphson method [36], the Fletcher-Powell method [32], [33], etc. IPM and the Newton-Raphson method are the most widely used since they have proven to be very efficient [37], [57].

In 1984, Karmarkar started to solve the LP problem via IPM, which is much more efficient than the traditional simplex algorithm, especially for large-scale problems. Traditionally, the simplex algorithm iterates among the vertices of the feasible region. Therefore, if the numbers of the variables and the constraints are both very large, the simplex algorithm is inefficient. Many iterations are required to reach the optimal vertex. IPM overcomes this drawback by traversing the interior of the feasible region. To maintain the feasibility, IPM transforms the problem to an unconstrained problem using barrier methods. The new objective function is formed by the sum of the equation constraints and the logarithmic barrier functions of the inequality constraints times respective multipliers. Then, Newton's method is used to solve the unconstrained problem. Transforming and solving steps continue iteratively until the solution converges.

In the early 1990s, IPM was introduced to solve OPF [57]. Several applications are listed here. First, IPM has proven to be attractive to deal with optimal reactive dispatch (ORD) for identifying active constraints intelligently and solving large-scale problems efficiently [27], [58]. However, since ORD is a highly nonlinear OPF problem with fixed real power injections and normally applied to networks under severe operating conditions, other transformation methods may cause severe numerical instabilities. Second, IPM is a good tool for the maximum loadability problem [3], where the objective function models the capabilities of load buses via scale factors, and the constraint set includes the power flow equations with these factors and the operating constraints. Third,

IPM is used for OPF with multiple objectives, where the objective function in IPM is a linear combination of all objectives with weighing factors. Fourth, IPM is a popular tool for solving the LP problem in successive linear programming algorithms [59]. Fifth, Torres and Quintana proposed an IPM-based OPF algorithm using voltage rectangular coordinates for better convergence, since the quadratic formulation leads to a constant the Hessian matrix [28]. In recent years, Jabr extended the quadratic model and used a scaling method in IPM [18].

IPM has several versions, such as the primal-dual [60] algorithm, the predictor-corrector [61] algorithm, and the multiple-centrality-corrections (MCC) [62] algorithm. The primal-dual algorithm performs a linear search within the feasible region. In the search space, the primal-dual algorithm determines the moving step and measures the desirability of each point. The origins of the searching directions are computed via Newton's method for the nonlinear equations. The predictor-corrector algorithm is a famous revision of IPM reported in [61]. In each iteration, the algorithm first estimates potential variable changes and then adjusts the estimation according to the values of the nonlinear terms. This method converges faster than primal-dual IPM since the quadratic items are included in the computation. The MCC algorithm needs a prediction step and a correction step and focuses on exploring matrix factorization. The prediction step is the same as in the predictor-corrector method but the correction step may have more than one term. The aims of correction are enlarging the step length of the current iteration, improving the centrality of the next iteration, and increasing the speed reaching feasibility.

Newton's method has proven to be the most efficient method in solving unconstrained optimization problems. Before Newton's method was used in OPF,

Dommel and Tinny solved OPF using the reduced gradient method with slow astringency in 1968 [11], and then Stott introduced secure constraints into their framework [19]. Later on, a quasi-Newton method was proven to have a super-linear convergence speed, much faster than the gradient method [13]. However, Newton's method may zigzag in some specific conditions [38], especially when approaching the optimal point, so some special strategies were designed when updating the variable values. For large-scale problems, Newton's method integrates some decomposition methods to reduce the size of the OPF problem [39]. Therefore, the Newton-based algorithm has been successfully applied to practical power systems with 1200 to 1500 buses [40].

Quadratic programming (QP) is a special case in nonlinear programming that includes only quadratic functions. In early years, Reid and Hasdorff formulated OPF as a QP problem using the Lagrange multiplier method and Taylor expansion [89] and solved QP via successive linear programming. Then, El-Kady et al. solved QP using Newton's method [42], and Tognola and Bacher designed a QP algorithm with quadratic convergence speed using the augmented Lagrangian method [43]. Moreover, the quadratic formulation of power systems can be integrated into quadratic programming without approximation [44].

2.3.2 Intelligent Search Algorithms

With the development of artificial intelligence, the intelligent search has become a very important technique in searching the global or near-global optimal solution. Main methods in this category are simulated annealing (SA), the genetic algorithm (GA), the evolution algorithm (EA), the particle swarm optimization (PSO), and artificial neural network (ANN). If the objective function is nonconvex, the solution may be trapped in a local optimum point. The random strategies in these intelligent search algorithms can help the solution jump out of the local optimum point. Different types of problems require different strategies. Researchers have already proposed several OPF algorithms

based on GA, EA, PSO, and ANN [63]. However, intelligent search methods are first-order methods and are inefficient for large-scale systems.

GA requires each possible OPF solution to be coded in chromosomes as a member of the population. Chromosomes in each generation are coded using a binary chain and ranked by a specific criterion. GA runs from one population to the next generation, and the members of the generation improve iteratively [31].

Bakirtzis et al. proposed the first EA-based OPF algorithm [64]. Then, Cai et al. solved the transient stability-constrained optimal power flow (TSCOPF) problem using modified EA, a differential evolution algorithm (DEA) with strong ability in searching for the global optimum. TSCOPF is a nonlinear optimization problem with both algebraic and differential equations. DEA solves it by employing both time-domain simulation and the transient energy function [29] due to DE's flexibility.

The idea of PSO [65] comes from the social behavior of organisms, such as fish schooling and birds flocking. PSO mimics the behaviors of looking for food, determining positions and the velocities of organisms back and forth in each iteration since the velocities in the next iteration can be represented by a random function of current positions and velocities. PSO will keep the best value through the current iteration and stop when meeting the preset criteria. Several PSO and modified PSO algorithms are proposed for OPF and SCOPF [66]-[70]. In addition, PSO algorithms have been applied to OPF with discrete control variables [30].

2.3.3 Sequential Algorithms

Sequential algorithms usually use LP or QP as tools to improve the solution in each iteration. Since Danzig proposed the simplex algorithm in 1947, LP has become a very important optimization tool. In the early years, Danzig and Wolfe's algorithm [51] and the revised simplex algorithm [52] were pioneering linear optimization methods for OPF. The cost function and the constraints are both linearized and solved via the simplex

algorithm or the primal-dual algorithm. Then, Stott and Hobson designed an iterative LP-based algorithm with a piecewise linear objective function for a power system with security constraints [12]. The violated constraint is added into the basis as an equation. This algorithm chooses the eligible row whose variable would result in the greatest reduction of the incoming constraint violation. Since the objective function is piecewise, the algorithm maintains optimality in every piece of linear section and finally reaches the global optimum when the solution becomes feasible. A similar algorithm using the revised simplex algorithm is then proposed in [53], where the objective function is also piecewise. Therefore, the solution will lie on segment boundaries. Segment breakpoints are determined according to the curve slope [2] and chosen before each iteration. Larger size means a worse solution, but smaller size means more iterations.

A typical QP has a quadratic objective function and linear constraints. Therefore, sequential algorithms can replace LP by QP especially when the original problem has a quadratic objective function. Solving methods to QP usually include IPM [124], the augmented Lagrangian method [125], the conjugate gradient method [126], the extended simplex method [127], etc.

Selecting the penalty function is a very popular topic in sequential algorithms. A well-designed penalty function can guarantee moving from infeasibility to feasibility since the power system is nonlinear and the initial working point may be infeasible [54]. If some constraints are violated initially, penalty factors can be attached to these constraints. As the penalty items reduce, system states get close to the feasible region [55]. The state moves along the gradient of the penalty function, and the step size is determined by infeasible variables and constraints [56]. If the state cannot move into the feasible region, the algorithm stops at an infeasible solution with some violated constraints. To enhance the usability of the solution of infeasible systems, the proposed algorithm returns an infeasible solution with mismatch residuals, and this solution can be

translated into a set of remedial actions that is the best possible solution for the problem at hand.

2.3.4 Decomposition Technologies

In recent years, researchers have extended OPF research to several advanced problems, such as SCOPF, multiple-objective OPF, a combination of OPF and unit commitment, etc. Because these problems usually have decomposable structures, researchers have proposed several decomposition methods [1], [45] where Benders' decomposition [47] is the most famous one. However, general OPF does not use decomposition methods since its problem structure may not be fit for decomposition.

For SCOPF, Benders' decomposition breaks down the original problem into one master problem and several slave problems. The master problem determines whether to connect devices, such as new generators, VAR sources, and capacitors [17], [21]. A slave problem is usually another OPF/SCOPF problem associated with one contingency. The master problem is the final solution when each slave problem is feasible [46]. The slave problem is solved via LP, and the master problem is solved via integer programming [48].

The problem combining both OPF and unit commitment can be decomposed into three levels: the mixed-integer linear decision problem, 24-hour nonlinear programming problems, and base OPF problems [49]. For multiple-objective OPF, the master problem is a global dispatch problem, and a slave problem is an OPF problem with weighted objective functions [4]. In addition, distributed algorithms and ordinal optimization (OO) are proposed to solve large-scale power systems, which can be viewed as distributed systems composed of several subsystems [23], [50], especially when the system has FACT devices enhancing the system security.

2.4 Power System Modeling

The complex power flow equations at each bus fully describe the configuration of a power system. Traditional power flow is a trigonometric form, also referred to as polar power flow, which is the most widely-used formulation, whether in power flow, state estimation, or OPF. The variables in polar power flow are the real and reactive powers of generators, the voltage magnitudes, the phase angles, the transformer tap settings, the capacitor bank status, etc. Polar power flow uses sine and cosine functions to describe the relationship between voltages and powers, so Newton's method takes more iterations when solving the power flow since the Hessians matrix is not constant [71].

To overcome the drawback of polar power flow, researchers designed a rectangular model to improve the convergence property of Newton's method [72]. Several works show its successful applications on the IPM-based OPF algorithms. In primal-dual IPM, Newton's method converges faster since the Jacobian matrix is constant only with a few exceptions. In predictor-corrector IPM, since the nonlinear terms of the power flow equations and the operating constraints are all quadratic, the corrector step estimates those nonlinear terms directly. Otherwise, obtaining the accurate values of higher-order terms [72] requires much more computing time. Jabr [18] proposed quadratic models for tap-changing transformers and unified power flow controller (UPFC) devices. He used primal-dual IPM and the same power flow model as those in [72].

The third type is quadratized power flow with only quadratic functions [73]. Different from previous works, quadratized power flow uses current balance instead of power balance at each bus. In this model, KCL describes system construction, and Thevenin's theorem gives the internal structures of generators and transformers. Since quadratized power flow includes the internal states of devices and all equations are quadratic, it has more equations, more state variables, and a larger Jacobian matrix. A larger Jacobian matrix means much more runtime in computing its inverse matrix, but this problem can be solved using sparsity techniques. Two successful works using

quadrated power flow have been reported, one for the steady-state and dynamic analysis of induction motors [74] and the other for contingency simulation [75].

2.5 Three-Phase Optimal Power Flow

Three-phase optimal power flow (TOPF) is a tool to find the optimum of a power system with distribution networks via managing controllable devices. After smart grid concepts were proposed, researchers paid more and more attention to issues on distribution grids, such as three-phase state estimation [91], [92] and TOPF [94]-[102]. Optimization on distribution grids previously focused on the reconfiguration of distribution systems for loss reduction [104]-[106] and sometimes service restoration [107]-[109]. Only a few research works solved distribution optimization via the OPF technology, referred to as TOPF. As more and more renewable sources are connected to distribution grids, distribution control is not limited to system reconfiguration. Controllable devices in distribution grids include generators, capacitor banks, shunt compensators, static VAR compensators (SVC), static synchronous compensators (STATCOM), plug-in hybrid electric vehicles (PHEV), storage systems, etc. In addition, there may be many more types of devices in the future. Therefore, TOPF is important and much more complicated than single-phase OPF.

Single-phase OPF algorithms model a power system using only one phase since they run on balanced transmission grids. However, TOPF algorithms are designed for unbalanced three-phase power systems with both transmission grids and distribution grids. In addition, TOPF has more types of integer variables, such as switches for capacitor banks and PHEVs. In the early 2000s, Hong and Wang proposed a TOPF method using Newton's method with SVCs for off-line use [94]. In recently years, researchers have published more TOPF works. Khodr et al. combined network reconfiguration with OPF via Benders' decomposition [96]. Zhu and Tomsovic's method uses the greedy algorithm and the steepest descent algorithm to dispatch small generators and storage resources in

distribution grids [97]. Harrison et al. proposed an approach to select optimal distributed generation (DG) sources via genetic algorithms and OPF [99]. Dolan et al. applied OPF to voltage control, power flow management, and restoration in active distribution networks [99]. Ahmadi et al. developed an OPF based algorithm to maximize real power outputs of DG sources for radial and meshed distribution networks [101]. Ochoa and Harrison's method focuses on minimizing energy loss with renewable distributed generation via multi-period AC OPF [97]. They also offered a method to evaluate the maximum capacity of new variable-distributed generation, which can be connected on a distribution network with active network management [94]. Bruno et al. proposed an unbalanced TOPF algorithm for on-line use in distribution management systems via Newton's method [102] with initial three-phase load flow given by OpenDSS [103].

2.6 Linearization Techniques

Early OPF works approximated the nonlinear optimization problem via piecewise linear functions when using LP methods [12]. However, piecewise linearization consumes plenty of computational resources since large-scale power systems have many equations and variables. In addition, determining the length of each linear segment is also difficult since the objective function and constraints are all nonlinear. The linear segments for the objective function may not fit for the constraints due to different nonlinearities. These problems limit the performance of piecewise linear algorithms.

Another linear approximation is formed via the linear combination of all control variables at the current working point. The coefficient of each control variable is its total derivative computed according to its definition, an infinitesimal change in the function with respect to the control variable. However, this method requires too much computation since the updated value of each function is obtained by solving the power flow, which is a very time-consuming task. Therefore, this is an inefficient linearization method.

The third method is the sensitivity method, also referred to as the co-state method, presented in [2]. This method returns the theoretical derivative values with respect to all control variables by the chain rule formula since the state variables are the functions of the control variables. The co-state method is efficient for LP-based algorithms since the inverse Jacobian matrix in the chain rule formula can be computed using sparsity techniques and can also be used in Newton's method. The proposed algorithm uses the co-state method with some modifications to accommodate the system model.

2.7 Summary

This chapter presented a comprehensive literature review of the research topic in this dissertation. This work studies the OPF problem on both single- and three-phase power systems. The areas of this research also include electric power system modeling, linearization techniques, and an OPF application: computing the operating cost in an optimal VAR allocation problem.

The OPF problem is well known to electric utilities and has been researched for several decades. People have developed many models, algorithms, and software packages. However, OPF is still a popular topic since it is the choice tool to provide the optimal solution to power systems. Researchers and engineers keep focusing on OPF problems since power systems are very complicated and continuously growing, and many new types of devices are developed and plugged in to the grid. Modeling power systems and their connected electric devices is a very important issue in OPF research. The form of the power flow equations is highly related to the convergence properties in solving the power flow, a major step in OPF. This work uses quadratized single- and three-phase power system models since quadratization exhibits fast convergence.

Existing OPF algorithms can be classified into three categories: nonlinear programming, intelligent search, and sequential algorithms. Nonlinear programming and sequential algorithms both perform well in large-scale power systems, while intelligent

search is inefficient. Nonlinear programming methods focus on the Kuhn-Tucker conditions. Sequential algorithms approach the optimal solution via moving the current working point according to optimization methods, such as LP or QP. However, all these algorithms require a feasible power flow solution as the starting point. To address this issue, this thesis work proposed a sequential algorithm starting from an arbitrary state possible infeasible and can therefore provide a solution for both feasible and infeasible power systems.

In addition, a method is proposed to solve the TOPF problem including both transmission grids and distribution grids. Several TOPF formulations and algorithms have been studied in the recent years, especially after the community started to pay attention to smart-grid technologies. However, no TOPF work has been done via quadratized three-phase power system modeling as in this dissertation. Finally, this work demonstrates an application to TOPF via an optimal VAR allocation problem.

CHAPTER 3

QUADRATIC POWER SYSTEM MODELING

3.1 Quadratic Power System Modeling

3.1.1 Overview

The traditional power flow formulation is based on the power balance equations and the use of polar coordinates for bus voltages. This formulation leads to a set of nonlinear equations that contain sine and cosine terms. A better approach, which introduces less complex and nonlinear equations, is known as quadratized power flow. Specifically, the quadratized power flow formulation is based on Kirchhoff's current law (nodal formulation) and Cartesian coordinates for bus voltages and the nonlinear models are converted to quadratic by the introduction of additional variables. The proposed OPF algorithm selects quadratized power flow, because quadratization limits the linearization error in the SLP method and Newton's method has quadratic convergence speed in solving the quadratized power flow. Although quadratized power flow includes more power flow equations which lead to a larger Jacobian matrix, this problem can be solved by sparsity technologies.

Quadratized power flow equations are sorted according to bus indices. The first two equations of each bus are the real and imaginary current conservation equations according to Kirchhoff's current law. They describe that the sum of the current flowing from each bus is zero. The remaining equations of each bus describe the internal states of some connected devices, such as synchronous generators, constant power loads, and single-phase transformers. Some internal state variables are introduced to form quadratic equations. In addition, Section 3.1.2 presents the equations of a synchronous generator

and a constant power load, and Appendix A shows the details of the quadratic single-phase transformer model.

3.1.2 Quadratic General Bus Modeling

3.1.2.1 Overview

Figure 3.1 describes a general bus with synchronous generators, mismatch current sources, constant power loads, constant impedance loads, capacitor banks, inductors, single-phase transformers, and circuit branches. Several constant power loads can be viewed as one device in the power flow. Constant impedance loads, capacitor banks, and inductors also hold this property.

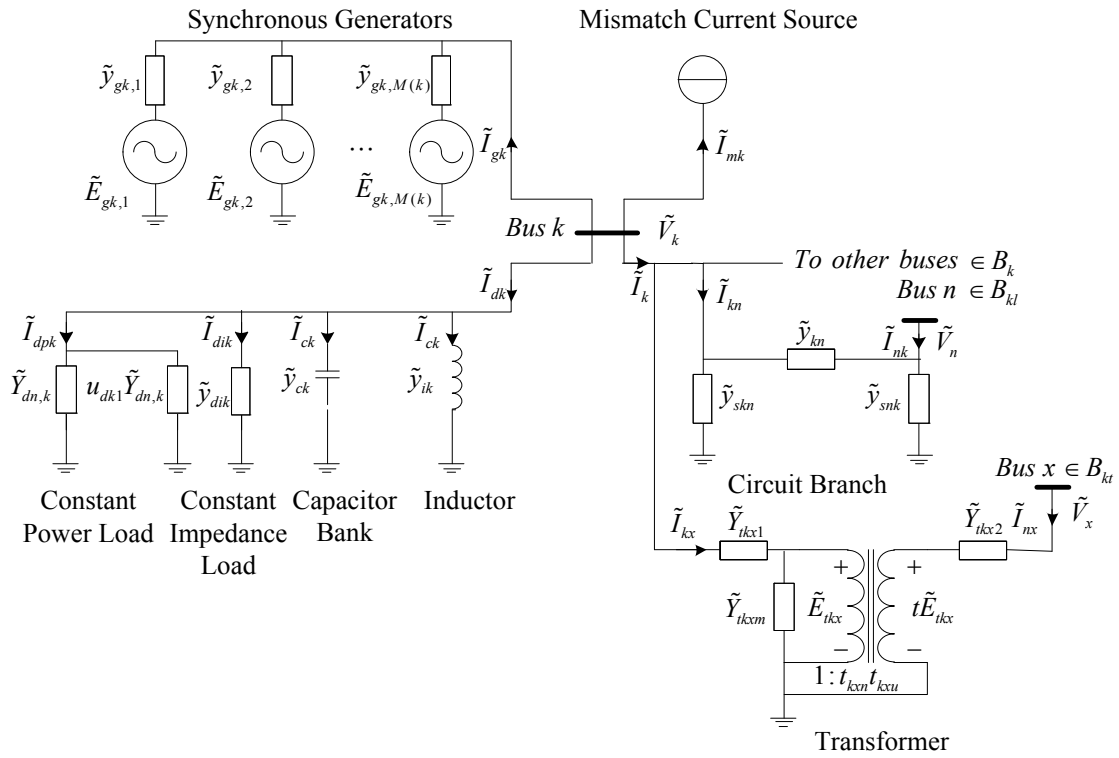


Figure 3.1 A general power system bus

In Figure 3.1, $M(k)$ is the number of the generators at Bus k . B_k is the index set of the buses adjacent to Bus k . B_{kl} is the index set of the buses connected to Bus k through a

transmission line. B_{kt} is the index set of the buses connected to Bus k through a transformer. It is known that $B_k = B_{kl} + B_{kt}$.

Mismatch current sources are artificial current sources retaining system status in the feasible region. The current flowing into each mismatch current source is \tilde{I}_{mk} , where $\tilde{I}_{mk} = -\tilde{I}_{gk} - \tilde{I}_{dk} - \tilde{I}_k$. \tilde{I}_{mk} is usually nonzero at the beginning of the algorithm and reduces iteratively. When \tilde{I}_{mk} equals zero, the algorithm reaches the optimal solution; otherwise, no feasible solution exists. \tilde{I}_{mk} provides the information for a set of remedial actions, such as load shedding. If the general bus is modeled using power conservation, the mismatch sources are real and reactive power sources shown in Section 8.2.1.

3.1.2.2 Frequency-Domain Model

The frequency-domain model at each bus consists of Kirchhoff's current law and the device equations of that bus. According to the current conservation at Bus k ,

$$0 = \tilde{I}_{gk} + \tilde{I}_{dk} + \tilde{I}_{mk} + \tilde{I}_k, \quad (3-1)$$

where $\tilde{I}_{dk} = \tilde{I}_{dpk} + \tilde{I}_{dik} + \tilde{I}_{ck} + \tilde{I}_{ik}$ and $\tilde{I}_k = \sum_{n \in B_{kl}} \tilde{I}_{kn} + \sum_{x \in B_{ktp}} \tilde{I}_{kx} + \sum_{x \in B_{kts}} \tilde{I}_{kx}$.

By defining $\tilde{y}_k = \tilde{y}_{dik} + \tilde{y}_{ck} + \tilde{y}_{ik}$ and substituting device parameters into (3-1), the frequency-domain model of the power system is

Bus:

$$\begin{aligned} 0 = & \sum_{j=1}^{M(k)} \left[\tilde{y}_{gk,j} (\tilde{V}_k - \tilde{E}_{gk,j}) \right] + (\tilde{Y}_{dn,k} \tilde{V}_k + u_{dk1} \tilde{Y}_{dn,k} \tilde{V}_k) + \tilde{y}_k \tilde{V}_k \\ & + \sum_{n \in B_{kl}} \left[(\tilde{y}_{kn} + \tilde{y}_{skn}) \tilde{V}_k - \tilde{y}_{kn} \tilde{V}_n \right] + \sum_{x \in B_{ktp}} \tilde{Y}_{tkx1} (\tilde{V}_k - \tilde{E}_{tkx}) \\ & + \sum_{x \in B_{kts}} \tilde{Y}_{txk2} (\tilde{V}_x - t_{xkn} t_{xku} \tilde{E}_{txk}) + \tilde{I}_{mk} \end{aligned} \quad (3-2)$$

Generators: (The generators are connected.)

$$P_{gk,j} + jQ_{gk,j} = -\tilde{y}_{gk,j}^* \tilde{V}_k (\tilde{V}_k - \tilde{E}_{gk,j})^* \quad (j=1,2,\dots,M(k)), \quad (3-3)$$

Constant power loads: (The constant power loads are connected.)

$$0 = (g_{dn,k} - jb_{dn,k})|\tilde{V}_k|^2 + u_{dk1}(g_{dn,k} - jb_{dn,k})|\tilde{V}_k|^2 - P_{dk} - jQ_{dk}, \text{ and} \quad (3-4)$$

Transformers: (The transformers are connected and Bus k is at the primary side.)

$$0 = -\tilde{Y}_{tkx1}(\tilde{V}_k - \tilde{E}_{tkx}) - t_{kxn}t_{kxu}Y_{tkx2}(\tilde{V}_x - t_{kxn}t_{kxu}\tilde{E}_{tkx}) + \tilde{Y}_{tkxm}\tilde{E}_{tkx}, \quad (3-5)$$

where B_{ktp} is the index set of the buses (secondary side) connected to Bus k (primary side) through a transformer, and B_{kts} is the index set of the buses (primary side) connected to Bus k (secondary side) through a transformer. The size of B_{ktp} is $T(k)$, and $B_{kt} = B_{ktp} \cup B_{kts}$.

3.1.2.3 Frequency-Domain Quadratic Model

Decomposed into real and imaginary parts, the frequency model forms the frequency-domain quadratic model: $\mathbf{g}_k(\mathbf{x}, \mathbf{u}, \mathbf{I}_m) = 0$. They are listed as follows:

Bus: ($\# = 3$)

$$\begin{aligned} \mathbf{g}_{ki} : 0 = & \left[\sum_{n \in B_{kl}} (b_{kn} + b_{skn}) + \sum_{j=1}^{M(k)} b_{gk,j} + \sum_{x \in B_{ktp}} (2t_{kxn}^2 b_{tkx}) + b_{dn,k} + b_{dn,k} u_{dn1} + b_k \right] V_{kr} \\ & + \left[\sum_{n \in B_{kl}} g_{kn} + \sum_{j=1}^{M(k)} g_{gk,j} + \sum_{x \in B_{ktp}} (2t_{kxn}^2 g_{tkx}) + g_{dn,k} + g_{dn,k} u_{dn1} + g_k \right] V_{ki} \\ & - \sum_{n \in B_{kl}} b_{kn} V_{jr} - \sum_{n \in B_{kl}} g_{kn} V_{ji} + \sum_{x \in B_{kts}} 2b_{txk} u_{txk2} V_{kr} + \sum_{x \in B_{kts}} 2g_{txk} u_{txk2} V_{ki} \\ & - \sum_{x \in B_{ktp}} 2t_{kxn}^2 b_{tkx} E_{tkxr} - \sum_{x \in B_{ktp}} 2t_{kxn}^2 g_{tkx} E_{tkxi} \\ & - \sum_{x \in B_{kts}} 2t_{xkn} b_{txk} u_{txk3} E_{txkr} - \sum_{x \in B_{kts}} 2t_{xkn} g_{txk} u_{txk3} E_{txki} \\ & - \sum_{j=1}^{M(k)} b_{gk,j} E_{gk,jr} - \sum_{j=1}^{M(k)} g_{gk,j} E_{gk,ji} + I_{mki} \end{aligned}, \quad (3-6)$$

$$\begin{aligned}
g_{kr} : 0 = & \left[\sum_{n \in B_{kl}} g_{kn} + \sum_{j=1}^{M(k)} g_{gk,j} + \sum_{x \in B_{kp}} (2t_{kxn}^2 g_{tkx}) + g_{dn,k} + g_{dn,k} u_{dk1} + g_k \right] V_{kr} \\
& - \left[\sum_{n \in B_{kl}} (b_{kn} + b_{skn}) + \sum_{j=1}^{M(k)} b_{gk,j} + \sum_{x \in B_{kp}} (2t_{kxn}^2 b_{tkx}) + b_{dn,k} + b_{dn,k} u_{dn1} + b_k \right] V_{ki} \\
& - \sum_{n \in B_{kl}} g_{kn} V_{jr} + \sum_{n \in B_k} b_{kn} V_{ji} + \sum_{x \in B_{ks}} 2g_{txk} u_{txk2} V_{kr} - \sum_{x \in B_{ks}} 2b_{txk} u_{txk2} V_{ki} \\
& - \sum_{x \in B_{kp}} 2t_{kxn}^2 g_{tkx} E_{tkxr} + \sum_{x \in B_{kp}} 2t_{kxn}^2 b_{tkx} E_{tkxi} \\
& - \sum_{x \in B_{ks}} 2t_{xkn} g_{txk} u_{txk3} E_{txkr} + \sum_{x \in B_{ks}} 2t_{xkn} b_{txk} u_{txk3} E_{txki} \\
& - \sum_{j=1}^{M(k)} g_{gk,j} E_{gk,jr} + \sum_{j=1}^{M(k)} b_{gk,j} E_{gk,ji} + I_{mkr}
\end{aligned} \tag{3-7}$$

$$g_{kV} : 0 = V_{kr}^2 + V_{ki}^2 - V_{knag}^2. \tag{3-8}$$

Generators: ($\# = 2M(k)$ if the generators are connected.)

$$\begin{aligned}
g_{gk,jQ} : 0 = & -b_{gk,j} V_{kr}^2 + b_{gk,j} V_{kr} E_{gk,jr} + g_{gk,j} V_{kr} E_{gk,ji} \\
& - b_{gk,j} V_{ki}^2 - g_{gk,j} V_{ki} E_{gk,jr} + b_{gk,j} V_{ki} E_{gk,ji} + Q_{gk,j} \quad (j=1, \dots, M(k))
\end{aligned} \tag{3-9}$$

$$\begin{aligned}
g_{gk,jP} : 0 = & g_{gk,j} V_{kr}^2 - g_{gk,j} V_{kr} E_{gk,jr} + b_{gk,j} V_{kr} E_{gk,ji} \\
& + g_{gk,j} V_{ki}^2 - b_{gk,j} V_{ki} E_{gk,jr} - g_{gk,j} V_{ki} E_{gk,ji} + P_{gk,j} \quad (j=1, \dots, M(k))
\end{aligned} \tag{3-10}$$

Constant Power Loads: ($\# = 2$ if the constant power loads are connected.)

If the real power output is nonzero,

$$g_{dkP} : 0 = g_{dn,k} u_{dk1} u_{dk2} + g_{dn,k} u_{dk2} - P_{dk}, \tag{3-11}$$

If the real power output is zero and the reactive power output is nonzero,

$$g_{dkQ} : 0 = b_{dn,k} u_{dk1} u_{dk2} + b_{dn,k} u_{dk2} + Q_{dk}, \tag{3-12}$$

$$g_{dkV} : 0 = V_{kr}^2 + V_{ki}^2 - u_{dk2}, \tag{3-13}$$

Transformers: ($\# = 6T(k)$ if the transformers are connected. Bus k is at the primary side.

Bus x is at the secondary side.)

$$\begin{aligned}
g_{tki} : 0 = & 2t_{kxn}^2 g_{tkx} V_{ki} + 2t_{kxn}^2 b_{tkx} V_{kr} + 2t_{kxn} g_{tkx} u_{tkx3} V_{xi} + 2t_{kxn} b_{tkx} u_{tkx3} V_{xr} \\
& - (2t_{kxn}^2 g_{tkx} + g_{tkxm}) E_{tkxi} - 2t_{kxn}^2 g_{tkx} u_{tkx4} E_{tkxi} - (2t_{kxn}^2 b_{tkx} + b_{tkxm}) E_{tkxr} - 2t_{kxn}^2 b_{tkx} u_{tkx4} E_{tkxr}
\end{aligned} \tag{3-14}$$

$$g_{tkr} : 0 = 2t_{kcn}^2 g_{tkx} V_{kr} - 2t_{kcn}^2 b_{tkx} V_{ki} + 2t_{kcn} g_{tkx} u_{tkx3} V_{xr} - 2t_{kcn} b_{tkx} u_{tkx3} V_{xi} - (2t_{kcn}^2 g_{tkx} + g_{tkcm}) E_{tkxr} - 2t_{kcn}^2 g_{tkx} u_{tkx4} E_{tkxr} + (2t_{kcn}^2 b_{tkx} + b_{tkcm}) E_{tkxi} + 2t_{kcn}^2 b_{tkx} u_{tkx4} E_{tkxi}, \quad (3-15)$$

$$g_{tk1} : 0 = 2t_{kxu} - t_{kxu}^2 + u_{tkx1}^2 - 1, \quad (3-16)$$

$$g_{tk2} : 0 = u_{tkx2} + u_{tkx1} u_{tkx2} - 1, \quad (3-17)$$

$$g_{tk3} : 0 = u_{tkx2} t_{kxu} - u_{tkx3}, \text{ and} \quad (3-18)$$

$$g_{tk4} : 0 = u_{tkx3} t_{kxu} - u_{tkx4}. \quad (3-19)$$

The number of the quadratized power flow equations of Bus k is $3+2M(k)+2+6T(k)$ if constant power loads are connected. The variable number of Bus k is $3+4M(k)+2+7T(k)$. They are $[V_{kr}, V_{ki}, V_{kmag}, E_{gk,jr}, E_{gk,ji}, P_{gk,j}, Q_{gk,j}, u_{dk1}, u_{dk2}, t_{kxu}, E_{tkxr}, E_{tkxi}, u_{tkx1}, u_{tkx2}, u_{tkx3}, u_{tkx4}]$. ($j = 1, \dots, M(k)$ and $x \in B_{ktp}$.)

3.1.2.4 Variable Classification

The variables are classified into the control variables and the state variables. The control variables are obtained in the optimization step and are assumed to be known in solving the power flow. The state variables are computed using the power flow equations where # the power flow equations = # the state variables. The selection of the control variables and the state variables is based on bus mode, including PQ mode, PV mode, and slack mode.

PQ buses are the most common buses in a power system. Their control variable set is $[P_{gk,j}, Q_{gk,j}, t_{kxu}]^T$ ($j = 1, \dots, M(k), x \in B_{ktp}$). PV buses are usually the buses with large reactive power generation or reactive power compensation. Their control variable set is $[V_{kmag}, P_{gk,1}, P_{gk,j}, Q_{gk,j}]^T$ ($j = 2, \dots, M(k)$). A system has a unique bus serving as the reference bus, sometimes with a zero voltage angle, referred to as the slack bus usually with frequency regulation power plant or with maximum adjacent buses. Its control variable set is $[V_{kr}, V_{ki}, P_{gk,j}, Q_{gk,j}, t_{kxu}]^T$ ($j = 2, \dots, M(k), x \in B_{ktp}$).

3.1.2.5 System-Level Equations

This algorithm uses two system-level equations instead of using all of the power flow equations in the optimization step since this will reduce the problem size and runtime dramatically. The real and reactive power balance equations can be used if the system does not consist of transformers. Otherwise, the algorithm will use the sum of real and imaginary current conservation equations of all buses since power balance equations with the transformer model are fourth-order equations. In this case, the real and reactive power balance equations can be used for verification.

Power balance equations denote that the total apparent power generated minus the total apparent power consumption is zero. That is,

$$0 = \sum_{k=1}^N \left(\sum_{j=1}^{M(k)} \tilde{S}_{gj} - \tilde{S}_{mk} - \tilde{S}_{dk} - \tilde{S}_{dik} - \tilde{S}_{dck} - \sum_{n \in B_{kl}} \tilde{S}_{kn} - \sum_{x \in B_{kt}} \tilde{S}_{tx} \right). \quad (3-20)$$

By Separating Equation (3-20) into real and reactive powers and substituting device parameters, the following equations are obtained:

$$\begin{aligned} P(\mathbf{x}, \mathbf{u}) : 0 = & \sum_{k=1}^N \sum_{j=1}^{M(k)} P_{gj,k} - \sum_{k=1}^N P_{dk} \\ & - \sum_{k=1}^N \left[\left(g_k + \sum_{n \in B_{kl}} g_{kn} + 2 \sum_{x \in B_{ktp}} t_{kxn}^2 g_{tkx} + 2 \sum_{x \in B_{kts}} u_{tkx2} g_{tkx} \right) (V_{kr}^2 + V_{ki}^2) \right] \\ & + 2 \sum_{k=1}^N \sum_{n \in B_{kl}} g_{kn} (V_{kr} V_{nr} + V_{ki} V_{ni}) \\ & + \sum_{k=1}^N \sum_{x \in B_{ktp}} \left[-(2t_{kxn}^2 g_{tkx} + g_{tkxm})(E_{tkxr}^2 + E_{tkxi}^2) + 4t_{kxn}^2 g_{tkx} (V_{kr} E_{tkxr} + V_{ki} E_{tkxi}) \right] \\ & + \sum_{k=1}^N \sum_{x \in B_{kts}} \left[-2t_{kxn}^2 g_{tkx} u_{tkx4} (E_{tkxr}^2 + E_{tkxi}^2) + 4t_{kxn} g_{tkx} u_{tkx3} (V_{kr} E_{tkxr} + V_{ki} E_{tkxi}) \right] \\ & - \sum_{k=1}^N (V_{kr} I_{mkr} + V_{ki} I_{mki}) \end{aligned} \quad \text{and} \quad (3-21)$$

$$\begin{aligned}
Q(\mathbf{x}, \mathbf{u}) : 0 = & \sum_{k=1}^N \sum_{j=1}^{M(k)} Q_{gj,k} - \sum_{k=1}^N Q_{dk} \\
& + \sum_{k=1}^N \left[\left(b_k + \sum_{n \in B_{kl}} (b_{kn} + b_{skn}) + 2 \sum_{x \in B_{ktp}} t_{kxn}^2 b_{tkx} + 2 \sum_{x \in B_{kts}} u_{tkx2} b_{tkx} \right) (V_{kr}^2 + V_{ki}^2) \right] \\
& - 2 \sum_{k=1}^N \sum_{n \in B_{kl}} b_{kn} (V_{kr} V_{nr} + V_{ki} V_{ni}) \\
& + \sum_{k=1}^N \sum_{x \in B_{ktp}} \left[(2t_{kxn}^2 b_{tkx} + b_{tkxn})(E_{tkxr}^2 + E_{tkxi}^2) - 4t_{kxn}^2 b_{tkx} (V_{kr} E_{tkxr} + V_{ki} E_{tkxi}) \right] \\
& + \sum_{k=1}^N \sum_{x \in B_{kts}} \left[2t_{kxn}^2 b_{tkx} u_{tkx4} (E_{tkxr}^2 + E_{tkxi}^2) - 4t_{kxn} b_{tkx} u_{tkx3} (V_{kr} E_{tkxr} + V_{ki} E_{tkxi}) \right] \\
& + \sum_{k=1}^N (V_{kr} I_{mki} - V_{ki} I_{mkr})
\end{aligned} \tag{3-22}$$

Real and imaginary current conservation equations are $\sum_{k=1}^N g_{kr} = 0$ and $\sum_{k=1}^N g_{ki} = 0$.

By representing them explicitly, the following equations are obtained:

$$\begin{aligned}
& I_r(\mathbf{x}, \mathbf{u}) : \\
0 = & \sum_{k=1}^N \left\{ \begin{aligned} & \left[\sum_{n \in B_{kl}} g_{kn} + \sum_{j=1}^{M(k)} g_{gk,j} + \sum_{x \in B_{ktp}} (2t_{kxn}^2 g_{tkx}) + g_{dn,k} + g_{dn,k} u_{dk1} + g_k \right] V_{kr} \\ & - \left[\sum_{n \in B_{kl}} (b_{kn} + b_{skn}) + \sum_{j=1}^{M(k)} b_{gk,j} + \sum_{x \in B_{ktp}} (2t_{kxn}^2 b_{tkx}) + b_{dn,k} + b_{dn,k} u_{dn1} + b_k \right] V_{ki} \\ & - \sum_{n \in B_{kl}} g_{kn} V_{jr} + \sum_{n \in B_k} b_{kn} V_{ji} + \sum_{x \in B_{kts}} 2g_{txk} u_{txk2} V_{kr} - \sum_{x \in B_{kts}} 2b_{txk} u_{txk2} V_{ki} \\ & - \sum_{x \in B_{ktp}} 2t_{kxn}^2 g_{tkx} E_{tkxr} + \sum_{x \in B_{ktp}} 2t_{kxn}^2 b_{tkx} E_{tkxi} \\ & - \sum_{x \in B_{kts}} 2t_{xkn} g_{txk} u_{txk3} E_{txkr} + \sum_{x \in B_{kts}} 2t_{xkn} b_{txk} u_{txk3} E_{txki} \\ & - \sum_{j=1}^{M(k)} g_{gk,j} E_{gk,jr} + \sum_{j=1}^{M(k)} b_{gk,j} E_{gk,ji} + I_{mkr} \end{aligned} \right\} \text{ and } \tag{3-23}
\end{aligned}$$

$I_i(\mathbf{x}, \mathbf{u}) :$

$$0 = \sum_{k=1}^N \left\{ \begin{aligned} & \left[\sum_{n \in B_{kl}} (b_{kn} + b_{skn}) + \sum_{j=1}^{M(k)} b_{gk,j} + \sum_{x \in B_{ktp}} (2t_{kxn}^2 b_{tkx}) + b_{dn,k} + b_{dn,k} u_{dn1} + b_k \right] V_{kr} \\ & + \left[\sum_{n \in B_{kl}} g_{kn} + \sum_{j=1}^{M(k)} g_{gk,j} + \sum_{x \in B_{ktp}} (2t_{kxn}^2 g_{tkx}) + g_{dn,k} + g_{dn,k} u_{dn1} + g_k \right] V_{ki} \\ & - \sum_{n \in B_{kl}} b_{kn} V_{jr} - \sum_{n \in B_{kl}} g_{kn} V_{ji} + \sum_{x \in B_{kts}} 2b_{txk} u_{txk2} V_{kr} + \sum_{x \in B_{kts}} 2g_{txk} u_{txk2} V_{ki} \\ & - \sum_{x \in B_{ktp}} 2t_{kxn}^2 b_{tkx} E_{tkxr} - \sum_{x \in B_{ktp}} 2t_{kxn}^2 g_{tkx} E_{tkxi} \\ & - \sum_{x \in B_{kts}} 2t_{xkn} b_{txk} u_{txk3} E_{txkr} - \sum_{x \in B_{kts}} 2t_{xkn} g_{txk} u_{txk3} E_{txki} \\ & - \sum_{j=1}^{M(k)} b_{gk,j} E_{gk,jr} - \sum_{j=1}^{M(k)} g_{gk,j} E_{gk,ji} + I_{mki} \end{aligned} \right\}. \quad (3-24)$$

3.2 Summary

This chapter provided the quadratic power system model used in the proposed OPF algorithm. At the beginning, a general bus was modeled in the complex form in the frequency domain followed by its quadratized model. The general bus model includes synchronous generators, constant power loads, constant impedance loads, capacitor banks, inductors, transformers, circuit branches, mismatch current sources, etc. Next, the variable classification according to bus mode was presented. Finally, this chapter gave several quadratic system-level equations, such as the power balance equations and the current conservation equations.

CHAPTER 4

OPTIMAL POWER FLOW PROBLEM DEFINITION

4.1 Introduction

This chapter describes a quadratic OPF formulation of power systems. Chapter 5 presents an algorithm to solve OPF iteratively using sequential methods.

4.2 The Quadratic Problem Formulation

The cost function of a power system is the mismatch penalty plus the sum of the quadratic cost functions of all generators. Mismatch variables are real and imaginary currents injecting to each bus in the quadratic power system model. Therefore, the nonlinear optimization problem is

$$\begin{aligned} \min \quad & J(\mathbf{x}, \mathbf{u}) = \mu \sum (|\mathbf{I}_m|) + \sum_{k=1}^N \sum_{j=1}^{M(k)} c_{k,j}(\mathbf{x}, \mathbf{u}) \\ \text{s.t.} \quad & \mathbf{g}(\mathbf{x}, \mathbf{u}, \mathbf{I}_m) = 0, \\ & \mathbf{h}^{\min} \leq \mathbf{h}(\mathbf{x}, \mathbf{u}) \leq \mathbf{h}^{\max} \\ & \mathbf{u}^{\min} \leq \mathbf{u} \leq \mathbf{u}^{\max} \end{aligned} \quad (4-1)$$

where

\mathbf{x} is the state variable vector,

\mathbf{u} is the control variable vector,

\mathbf{I}_m is the mismatch current vector,

N is the total number of buses in the power system,

$M(k)$ is the number of generators at Bus k ,

$J(\mathbf{x}, \mathbf{u})$ is the objective function with the operation cost of the system and mismatch penalties taken into account,

$c_{k,j}(\mathbf{x}, \mathbf{u})$ is the cost function of the j th generator at Bus k , e.g., $c_{k,j}(\mathbf{x}, \mathbf{u}) = a_{k,j} + b_{k,j}P_{k,j} + c_{k,j}P_{k,j}^2$, where the unit of $a_{k,j}$, $b_{k,j}$, and $c_{k,j}$ are $\$/hour$, $\$/(MW \cdot hour)$, and $\$/(MW^2 \cdot hour)$ respectively,

$\mathbf{g}(\mathbf{x}, \mathbf{u}, \mathbf{I}_m) = 0$ are quadratized power flow equations, represented in Section 3.1.2.3,

$\mathbf{h}^{\min} \leq \mathbf{h}(\mathbf{x}, \mathbf{u}) \leq \mathbf{h}^{\max}$ are operating constraints, and

$\mathbf{u}^{\min} \leq \mathbf{u} \leq \mathbf{u}^{\max}$ are control variable constraints, upper and lower bounds of all control variables.

The variables are sorted according to the bus indices. The variables of Bus k include

Real and imaginary voltages, V_{kr} and V_{ki} ,

Real and reactive mismatch currents I_{mkr} and I_{mki} ,

Generator real and reactive powers $P_{gk,j}$ and $Q_{gk,j}$ (The generators are connected.),

Generator internal electromotive forces $E_{gk,jr}$ and $E_{gk,ji}$ (The generators are connected.),

Constant power load variables u_{dk1} and u_{dk2} (The constant power loads are connected.), and

Transformer variables t_{kxu} , E_{tkxr} , E_{tkxi} , u_{tkx1} , u_{tkx2} , u_{tkx3} and u_{tkx4} . (The transformers are connected.)

Operating constraints include

$$\left(V_{kmag}^{\min}\right)^2 \leq V_{kr}^2 + V_{ki}^2 \leq \left(V_{kmag}^{\max}\right)^2, \quad (4-2)$$

where V_{kmag} is the voltage magnitude at Bus k ,

$$P_{1,1}^{\min} \leq P_{1,1} \leq P_{1,1}^{\max}, \quad (4-3)$$

where $P_{1,1}$ is the real power of the slack mode generator,

$$Q_{k,j}^{\min} \leq Q_{k,j} \leq Q_{k,j}^{\max}, \quad (4-4)$$

where $Q_{k,j}$ is the reactive power of the slack mode generator or PV mode generators,

$$\sqrt{g_{kn}^2 + b_{kn}^2} \left(V_{kr}^2 + V_{ki}^2 + V_{nr}^2 + V_{ni}^2 - 2V_{kr}V_{nr} - 2V_{ki}V_{ni} \right) \leq S_{a,kn,\max}, \quad (4-5)$$

where S_{kn} is the apparent power transmission through a transmission line between the adjacent buses (Bus k and Bus n),

$$2t_{kn}^2 \sqrt{g_{tk}^2 + b_{tk}^2} \left(\begin{aligned} &V_{kr}^2 + V_{ki}^2 + V_{xr}^2 + V_{xi}^2 + E_{tkr}^2 + E_{tki}^2 - 2V_{kr}V_{xr} \\ &- 2V_{ki}V_{xi} - 2V_{kr}E_{tkr} - 2V_{ki}E_{tki} + 2V_{xr}E_{tkr} + 2V_{xi}E_{tki} \end{aligned} \right) \leq S_{a,tkx,\max}, \quad (4-6)$$

where S_{tkx} is the apparent power transmission through a transformer between adjacent buses (Bus k and Bus x).

Control variable constraints include the real voltage at the slack bus ($V_{1mag}^{\min} \leq V_{1r} \leq V_{1mag}^{\max}$) and transformer taps ($t_{kxu}^{\min} \leq t_{kxu} \leq t_{kxu}^{\max}$).

The slack bus is an arbitrary bus with generators connected. In order to facilitate notation and symbolism, it is assumed that the slack bus is Bus 1 and its first generator runs at slack mode. All other buses are PQ buses or PV buses. In a power system with N buses and N_p constant power loads, the total number of the variables is $\sum_{k=1}^N [4M(k) + 7T(k)] + 2N_p + 2N$. A transformation is introduced to reduce the number of the mismatch variables:

$$I_{mkr} = (1 - v)I_{mkr}^o \quad (v \in [0, 1]) \text{ and} \quad (4-7)$$

$$I_{mki} = (1 - v)I_{mki}^o \quad (v \in [0, 1]). \quad (4-8)$$

After the transformation, the mismatch variables \mathbf{I}_m are eliminated, and the total number of the variables reduces to $\sum_{k=1}^N [4M(k) + 7T(k)] + 2N_p + 1$. The variables of Bus k follow the rules:

1. v is selected to be a control variable for the system,
2. The number of \mathbf{x} of Bus k is $3 + 2M(k) + 2 + T(k)$, (The constant power loads are connected.)
3. The number of \mathbf{u} of Bus k is $2M(k) + 6T(k)$.

Therefore, the total variable number in \mathbf{x} is $\sum_{k=1}^N [2M(k) + 6T(k)] + 2N_p$, and the total number variable in \mathbf{u} is $\sum_{k=1}^N [2M(k) + T(k)]$ after the transformation.

4.3 Summary

This section presented the definition of the OPF problem in the quadratic formulation including linear and quadratic formulas only. The objective function of the OPF problem equals the mismatch penalties plus the fuel costs. Moreover, the constraint set includes the power flow equations, the operating constraints, and the control variable constraints. Detailed information about how to form the objective function and the constraints are also provided.

CHAPTER 5

THE PROPOSED OPTIMAL POWER FLOW ALGORITHM

5.1 Introduction

This chapter presents a sequential OPF algorithm with two implementation methods: sequential linear programming (SLP) and sequential quadratic programming (SQP). While these two implementation methods are similar, SLP linearizes all the functions and SQP keeps the objective function quadratic in the optimization step.

5.2 Algorithm Outline

The proposed algorithm starts from an infeasible optimal state of the system and maintains the current balance at each bus by introducing an artificial mismatch current source. The real and imaginary currents from this mismatch current source form mismatch variables. These mismatch variables reduce as iterations progress by introducing a unified control variable. If all the mismatch variables reach zero, the solution enters the feasible region and is optimal automatically. Otherwise, the algorithm provides a suboptimal solution with mismatch residuals. These residuals represent the system limitation and can be eliminated by load shedding. Figure 5.1 shows the flow chart of the algorithm using the SLP implementation.

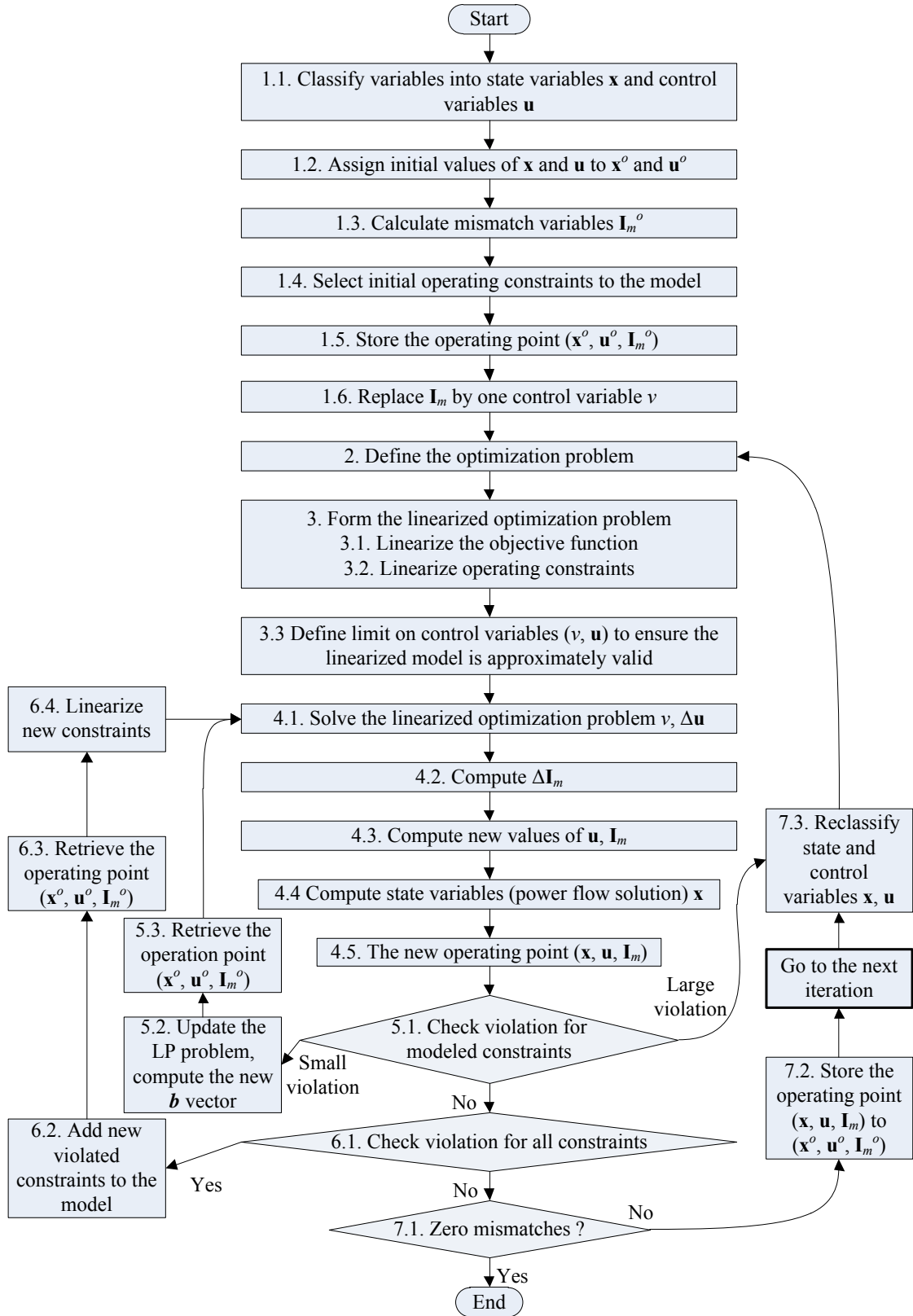


Figure 5.1 The flow chart of the SLP algorithm

Using the co-state method in each iteration, the algorithm first converts the OPF problem to a linearized optimization problem with the control variables only. The constraints consist of the real and imaginary power balance equations and the operating constraints violated in the previous iterations. The control variables are limited by their physical bounds and linearization limits computed according to the linearization error of the current balance equations. The algorithm then obtains the updated values of the control variables using LP or QP algorithms and the state variables by solving the power flow. If some modeled operation constraints are violated, the \mathbf{b} vector will be updated in the linearized optimization problem and the previous solution is retrieved. If some other constraints are violated, the algorithm adds these constraints, retrieves the previous solution, and linearizes new constraints. If the mismatch variables are nonzero, the next iteration starts and the variables may be reclassified.

5.3 SLP Algorithm Implementation

This section presents the detailed description of the SLP implementation. The SQP implementation will be shown in Section 5.4.

5.3.1 Initialization

5.3.1.1 Classify Variables into State Variables \mathbf{x} and Control Variables \mathbf{u}

A power system usually has three types of bus mode: slack mode, PQ mode, and PV mode. The classification of the control variables and the state variables is listed as follows:

Slack mode:

$$\mathbf{x} = [V_{1mag}, E_{g1jr}, E_{g1ji}, P_{g1,1}, Q_{g1,1}, u_{d11}, u_{d12}, E_{tkxr}, E_{tkxi}, u_{tkx1}, u_{tkx2}, u_{tkx3}, u_{tkx4}]$$

$$(j = 1, \dots, M(1), x \in B_{ktp}).$$

$$\mathbf{u} = [V_{1r}, V_{1i}, P_{g1j}, Q_{g1j}, t_{kxu}] (j = 2, \dots, M(1), x \in B_{ktp}).$$

PQ mode:

$$\mathbf{x} = [V_{kmag}, V_{kr}, V_{ki}, E_{gk,jr}, E_{gk,ji}, u_{dk1}, u_{dk2}, E_{tkxr}, E_{tkxi}, u_{tkx1}, u_{tkx2}, u_{tkx3}, u_{tkx4}]$$

$$(j = 1, \dots, M(k), x \in B_{ktp}).$$

$$\mathbf{u} = [P_{gk,j}, Q_{gk,j}, t_{kxu}] (j = 2, \dots, M(k), x \in B_{ktp}).$$

PV mode:

$$\mathbf{x} = [V_{kr}, V_{ki}, E_{gk,jr}, E_{gk,ji}, Q_{gk,1}, u_{dk1}, u_{dk2}, E_{tkxr}, E_{tkxi}, u_{tkx1}, u_{tkx2}, u_{tkx3}, u_{tkx4}]$$

$$(j = 1, \dots, M(k), x \in B_{ktp}).$$

$$\mathbf{u} = [V_{kmag}, P_{gk,1}, P_{gk,j}, Q_{gk,j}, t_{kxu}] (j = 2, \dots, M(k), x \in B_{ktp}).$$

where k is the index of each bus. Note: at the slack bus, one pair of power variables are control variables, and real and imaginary voltage variables are state variables.

5.3.1.2 Assign Initial Values of \mathbf{x} and \mathbf{u} to \mathbf{x}^o and \mathbf{u}^o

The power transmissions between buses are enforced to be zero to avoid violated transmission constraints when the algorithm begins. Therefore, the real and imaginary voltages at all buses are set to 1.0 and 0.0 respectively. Other control variables and state variables are set to some certain values within their physical bounds and the mismatch variables are calculated according to the power flow equations.

The initial variable values at Bus k are set as follows:

Bus voltages: $V_{kr} = 1.0$, $V_{ki} = 0.0$, and $V_{kmag} = 1.0$.

Generator powers: $P_{gk,j}$ is a valid value, where $P_{gk,j}^{\min} \leq P_{gk,j} \leq P_{gk,j}^{\max}$ ($j = 1, \dots, M(k)$), and

$Q_{gk,j}$ is a valid value, where $Q_{gk,j}^{\min} \leq Q_{gk,j} \leq Q_{gk,j}^{\max}$ ($j = 1, \dots, M(k)$).

The internal electromotive force of the j th generator can be derived from Equation (3-9) and Equation (3-10):

$$E_{gk,jr} = \frac{b_2 A - a_2 B}{a_2 b_1 - a_1 b_2} \text{ and} \quad (5-1)$$

$$E_{gk,ji} = \frac{b_1 A - a_1 B}{a_1 b_2 - a_2 b_1}, \quad (5-2)$$

where

$$a_1 = g_{gk,j} V_{kr} + b_{gk,j} V_{ki},$$

$$a_2 = -b_{gk,j} V_{kr} + g_{gk,j} V_{ki},$$

$$b_1 = b_{gk,j} V_{kr} - g_{gk,j} V_{ki},$$

$$b_2 = g_{gk,j} V_{kr} + b_{gk,j} V_{ki},$$

$$A = g_{gk,j} V_{kr}^2 + g_{gk,j} V_{ki}^2 + P_{gk,j}, \text{ and}$$

$$B = b_{gk,j} V_{kr}^2 + b_{gk,j} V_{ki}^2 - Q_{gk,j}.$$

Load variables: $u_{dk1} = 0$ and $u_{dk2} = V_{kmag}^2$.

The initial variable values of the transformers are $t_{kxu} = 1.03$, $u_{tk1} = 0.03$, $u_{tk2} = 0.9708738$, $u_{tk3} = 1.0$, and $u_{tk4} = 1.03$.

The internal electromotive force of the transformer connecting Bus k (primary side) and Bus x (secondary side) can be derived from Equation (3-14) and Equation (3-15):

$$E_{tkxr} = \frac{b_2 A - a_2 B}{a_2 b_1 - a_1 b_2} \text{ and} \quad (5-3)$$

$$E_{tkxi} = \frac{b_1 A - a_1 B}{a_1 b_2 - a_2 b_1}, \quad (5-4)$$

where

$$a_1 = -2t_{kxn}^2 g_{tkx} - g_{tkxm} - 2t_{kxn}^2 g_{tkx} u_{tkx4},$$

$$a_2 = 2t_{kxn}^2 b_{tkx} + b_{tkxm} + 2t_{kxn}^2 b_{tkx} u_{tkx4},$$

$$b_1 = -2t_{kxn}^2 b_{tkx} - b_{tkxm} - 2t_{kxn}^2 b_{tkx} u_{tkx4},$$

$$b_2 = -2t_{kxn}^2 g_{tkx} - g_{tkxm} - 2t_{kxn}^2 g_{tkx} u_{tkx4},$$

$$A = 2t_{kxn}^2 g_{tkx} V_{kr} - 2t_{kxn}^2 b_{tkx} V_{ki} + 2t_{kxn} g_{tkx} u_{tkx3} V_{xr} - 2t_{kxn} b_{tkx} u_{tkx3} V_{xi}, \text{ and}$$

$$B = 2t_{kcn}^2 g_{tkx} V_{ki} + 2t_{kcn}^2 b_{tkx} V_{kr} + 2t_{kcn} g_{tkx} u_{tkx3} V_{xi} + 2t_{kcn} b_{tkx} u_{tkx3} V_{xr}.$$

The initial variable values can be set to other values if this initial system state is feasible. Sometimes this setup can increase the convergence speed since the initial working point may be closer to the optimal solution. For example, in the IEEE test cases shown in Chapter 8, the initial variable values can be set to the default variable values instead of using the proposed initialization method before.

5.3.1.3 Calculate Mismatch Variables \mathbf{I}_m^o

The mismatch values are calculated using the power flow equations: g_{kr} and g_{ki} .

5.3.1.4 Select Initial Operating constraints to the Model

The initial operating constraints include the real and imaginary current conservation equations ($I_r(\mathbf{x}, \mathbf{u})$ and $I_i(\mathbf{x}, \mathbf{u})$) and the real and reactive power constraints of the slack generator.

5.3.1.5 Store Operating Point $(\mathbf{x}^o, \mathbf{u}^o, \mathbf{I}_m^o)$

That is $\mathbf{x}^o = \mathbf{x}$, $\mathbf{u}^o = \mathbf{u}$, and $\mathbf{I}_m^o = \mathbf{I}_m$.

5.3.1.6 Replace \mathbf{I}_m by One Control Variable v

To reduce the variable number in the optimization problem, the mismatch variables are substituted with one control variable v representing the normal change of the mismatch variables. Therefore, all the mismatch variables (a total of $2N$) are replaced by a single variable v and the optimization problem is converted to

$$\begin{aligned} \min \quad & f(\mathbf{x}, \mathbf{u}) = \mu(1-v) \sum \left(\left| \mathbf{I}_m^o \right| \right) + \sum_{k=1}^N \sum_{j=1}^l c_{k,j}(\mathbf{x}, \mathbf{u}) \\ \text{s.t.} \quad & \mathbf{g}(\mathbf{x}, \mathbf{u}, v) = 0 \\ & \mathbf{h}^{\min} \leq \mathbf{h}(\mathbf{x}, \mathbf{u}) \leq \mathbf{h}^{\max}, \\ & \mathbf{u}^{\min} \leq \mathbf{u} \leq \mathbf{u}^{\max} \\ & 0 \leq v \leq 1 \end{aligned} \tag{5-5}$$

where \mathbf{I}_m^o is the vector restoring the present values of the mismatch currents. The algorithm initializes v to 0 and sets that $\mathbf{I}_m = (1-v)\mathbf{I}_m^o$ ($v \in [0,1]$) before solving each converted problem. An upper bound is assigned to v to control the nonlinearity caused by mismatch changes. Once v reaches 1, the feasible and optimal solution is achieved.

5.3.2 Define the Optimization Problem

This subsection presents the nonlinear optimization problem simplified from the original OPF problem. This problem uses the real and imaginary current balance equations instead of the power flow equations. Since there are a large number of the power flow equations, LP runtime will highly decrease if those equations are removed. In addition, the operating constraints are excluded at the beginning and will be added adaptively since only a small part of them will be active in the end. This nonlinear optimization problem is defined as follows:

$$\begin{aligned}
\min \quad & \mu(1-v) \sum \left| \mathbf{I}_m^o \right| + \sum_{k=1}^N \sum_{j=1}^{M(k)} c_{k,j}(\mathbf{x}, \mathbf{u}) \\
s.t. \quad & I_r(\mathbf{x}, \mathbf{u}, v) = 0 \\
& I_i(\mathbf{x}, \mathbf{u}, v) = 0 \\
& \mathbf{h}^{\min} \leq \mathbf{h}(\mathbf{x}, \mathbf{u}) \leq \mathbf{h}^{\max} \\
& \mathbf{u}^{\min} \leq \mathbf{u} \leq \mathbf{u}^{\max}
\end{aligned} \tag{5-6}$$

where $I_r(\mathbf{x}, \mathbf{u}, v) = 0$ and $I_i(\mathbf{x}, \mathbf{u}, v) = 0$ are the real and imaginary current balance equations respectively. $\mathbf{h}^{\min} \leq \mathbf{h}(\mathbf{x}, \mathbf{u}) \leq \mathbf{h}^{\max}$ is the set of the operating constraints in the present iteration.

5.3.3 Form the Linearized Optimization Problem

The algorithm then eliminates the state variables defined in the problem above and the problem is re-casted in terms of only the control variables. This is achieved by linearization whereby all functions and quantities are expressed as the linear combinations of the control variables. Appendix B.3 shows the discussion of linearization.

The following subsections present the formulas to linearize the objective function and the operating constraints.

5.3.3.1 Linearize the Objective Function

The linearized form of $c_{k,j}(\mathbf{x}, \mathbf{u})$ (k or $j \neq 1$) is

$$c_{k,j}(\mathbf{x}, \mathbf{u}) = c_{k,j}(\mathbf{x}^o, \mathbf{u}^o) + \left. \frac{\partial c_{k,j}(\mathbf{x}^o, \mathbf{u}^o)}{\partial \mathbf{u}} \right|_{\mathbf{g}_{reduced}(\mathbf{x}, \mathbf{u}, \mathbf{v})=0} \cdot \Delta \mathbf{u} + o(\Delta \mathbf{u}), \quad (5-7)$$

where

$$\left. \frac{\partial c_{k,j}(\mathbf{x}^o, \mathbf{u}^o)}{\partial \mathbf{u}} \right|_{\mathbf{g}_{reduced}(\mathbf{x}, \mathbf{u}, \mathbf{v})=0} = \frac{\partial c_{k,j}(\mathbf{x}^o, \mathbf{u}^o)}{\partial \mathbf{u}} - \frac{\partial c_{k,j}(\mathbf{x}^o, \mathbf{u}^o)}{\partial \mathbf{x}} \left(\frac{\partial \mathbf{g}_{reduced}(\mathbf{x}^o, \mathbf{u}^o)}{\partial \mathbf{x}} \right)^{-1} \frac{\partial \mathbf{g}_{reduced}(\mathbf{x}^o, \mathbf{u}^o)}{\partial \mathbf{u}}.$$

5.3.3.2 Linearize the Operating Constraints

The linearized form of $h(\mathbf{x}, \mathbf{u})$ is

$$h(\mathbf{x}, \mathbf{u}) = h(\mathbf{x}^o, \mathbf{u}^o) + \left. \frac{\partial h(\mathbf{x}^o, \mathbf{u}^o)}{\partial \mathbf{u}} \right|_{\mathbf{g}_{reduced}(\mathbf{x}, \mathbf{u}, \mathbf{v})=0} \cdot \Delta \mathbf{u} + \left. \frac{\partial h(\mathbf{x}^o, \mathbf{u}^o)}{\partial \mathbf{v}} \right|_{\mathbf{g}_{reduced}(\mathbf{x}, \mathbf{u}, \mathbf{v})=0} \cdot \mathbf{v} + o(\Delta \mathbf{u}) + o(\mathbf{v}), \quad (5-8)$$

where

$$\left. \frac{\partial h(\mathbf{x}^o, \mathbf{u}^o)}{\partial \mathbf{u}} \right|_{\mathbf{g}_{reduced}(\mathbf{x}, \mathbf{u}, \mathbf{v})=0} = \frac{\partial h(\mathbf{x}^o, \mathbf{u}^o)}{\partial \mathbf{u}} - \frac{\partial h(\mathbf{x}^o, \mathbf{u}^o)}{\partial \mathbf{x}} \left(\frac{\partial \mathbf{g}_{reduced}(\mathbf{x}^o, \mathbf{u}^o, 0)}{\partial \mathbf{x}} \right)^{-1} \frac{\partial \mathbf{g}_{reduced}(\mathbf{x}^o, \mathbf{u}^o, 0)}{\partial \mathbf{u}} \text{ and}$$

$$\left. \frac{\partial h(\mathbf{x}^o, \mathbf{u}^o)}{\partial \mathbf{v}} \right|_{\mathbf{g}_{reduced}(\mathbf{x}, \mathbf{u}, \mathbf{v})=0} = \frac{\partial h(\mathbf{x}^o, \mathbf{u}^o)}{\partial \mathbf{v}} - \frac{\partial h(\mathbf{x}^o, \mathbf{u}^o)}{\partial \mathbf{x}} \left(\frac{\partial \mathbf{g}_{reduced}(\mathbf{x}^o, \mathbf{u}^o, 0)}{\partial \mathbf{x}} \right)^{-1} \frac{\partial \mathbf{g}_{reduced}(\mathbf{x}^o, \mathbf{u}^o, 0)}{\partial \mathbf{v}}.$$

The linearized optimization problem is obtained via substituting these equations into the problem in Section 5.3.2 and ignoring the higher order items.

5.3.3.3 The Linearized Optimization Problem

$$\begin{aligned}
\min \quad & \mu(1-v) \sum_{k=1}^N |\mathbf{I}_m^o| + \sum_{k=1}^N \sum_{j=1}^{M(k)} \left[c_{k,j}(\mathbf{x}^o, \mathbf{u}^o) + \frac{\partial c_{k,j}(\mathbf{x}^o, \mathbf{u}^o)}{\partial \mathbf{u}} \bigg|_{\mathbf{g}_{reduced}(\mathbf{x}, \mathbf{u}, v)=0} \cdot \Delta \mathbf{u} \right] \quad (k \text{ or } j \neq 1) \\
s.t. \quad & I_r(\mathbf{x}^o, \mathbf{u}^o, 0) + \frac{\partial I_r(\mathbf{x}^o, \mathbf{u}^o, 0)}{\partial v} \bigg|_{\mathbf{g}_{reduced}(\mathbf{x}, \mathbf{u}, v)=0} \cdot v + \frac{\partial I_r(\mathbf{x}^o, \mathbf{u}^o, 0)}{\partial \mathbf{u}} \bigg|_{\mathbf{g}_{reduced}(\mathbf{x}, \mathbf{u}, v)=0} \cdot \Delta \mathbf{u} = 0 \\
& I_i(\mathbf{x}^o, \mathbf{u}^o, 0) + \frac{\partial I_i(\mathbf{x}^o, \mathbf{u}^o, 0)}{\partial v} \bigg|_{\mathbf{g}_{reduced}(\mathbf{x}, \mathbf{u}, v)=0} \cdot v + \frac{\partial I_i(\mathbf{x}^o, \mathbf{u}^o, 0)}{\partial \mathbf{u}} \bigg|_{\mathbf{g}_{reduced}(\mathbf{x}, \mathbf{u}, v)=0} \cdot \Delta \mathbf{u} = 0 \quad , \\
& \Delta \mathbf{h}^{\min} \leq \frac{\partial \mathbf{h}(\mathbf{x}^o, \mathbf{u}^o)}{\partial v} \bigg|_{\mathbf{g}_{reduced}(\mathbf{x}, \mathbf{u}, v)=0} \cdot v + \frac{\partial \mathbf{h}(\mathbf{x}^o, \mathbf{u}^o)}{\partial \mathbf{u}} \bigg|_{\mathbf{g}_{reduced}(\mathbf{x}, \mathbf{u}, v)=0} \cdot \Delta \mathbf{u} \leq \Delta \mathbf{h}^{\max} \\
& \Delta \mathbf{u}^{\min} \leq \Delta \mathbf{u} \leq \Delta \mathbf{u}^{\max} \\
& 0 \leq v \leq v^{\max}
\end{aligned} \tag{5-9}$$

where $\Delta \mathbf{h}^{\min} = \mathbf{h}^{\min} - \mathbf{h}(\mathbf{x}^o, \mathbf{u}^o)$ and $\Delta \mathbf{h}^{\max} = \mathbf{h}^{\max} - \mathbf{h}(\mathbf{x}^o, \mathbf{u}^o)$ are the lower and upper bounds of the operating constraints in the linearized problem. The lower and upper bounds of the control variables in $\Delta \mathbf{u}^{\min}$ and $\Delta \mathbf{u}^{\max}$ are determined by linearization limit strategies shown in the next step.

5.3.3.4 Select Limits on Control Variables (v, \mathbf{u}) to Ensure Linearized Model is

Approximately Valid

The mismatch variables are set to reduce linearly, so the algorithm selects

$$v^{\max} = \frac{1.0}{Step_{mismatch} - Step_{now} + 1}, \tag{5-10}$$

where $Step_{mismatch}$ is the iteration number that reduces the mismatch to zero and $Step_{now}$ is the index of the current iteration.

The linearization limits of the other control variables are determined according to the linearization error since a larger error may cause an invalid system status. The linearization error is much larger for large-scale systems since it increases with the number of variables. This algorithm considers the effect of the linearization error on the

current conservation equations only, because these two equations guarantee a solution for the system and are very time efficient in computing the linearization errors. Furthermore, computing the linearization errors based on inequalities needs too much runtime. For example, the upper bound of V_{kmax} is represented by a function with state variables. The accurate value of V_{kmax} after changing a control variable must be computed via solving the power flow equations. Therefore, when the allowed error is η , the lower limits on u_i ($u_i \in \mathbf{u}, u_i \neq v$) is

$$\Delta u_i^{\min} = \max \left(u_i^{\min} - u_i^o, -\eta / \frac{dI_r(\mathbf{x}, \mathbf{u}, v)}{du_i}, -\eta / \frac{dI_i(\mathbf{x}, \mathbf{u}, v)}{du_i} \right) \text{ and} \quad (5-11)$$

the upper limits on u_i ($u_i \in \mathbf{u}, u_i \neq v$) is

$$\Delta u_i^{\max} = \min \left(u_i^{\max} - u_i^o, \eta / \frac{dI_r(\mathbf{x}, \mathbf{u}, v)}{du_i}, \eta / \frac{dI_i(\mathbf{x}, \mathbf{u}, v)}{du_i} \right), \quad (5-12)$$

where η is set according to system topology and parameters. For example, $\eta = 0.07$ for the RTS-79 system in Section 8.3.

5.3.4 Solve the System

5.3.4.1 Solve the Linearized Optimization Problem $v, \Delta \mathbf{u}$

The solution of linearized optimization problem gives v and $\Delta \mathbf{u}$.

5.3.4.2 Compute $\Delta \mathbf{I}_m$

The mismatch change $\Delta \mathbf{I}_m = -v \mathbf{I}_m^o$.

5.3.4.3 Compute the New Values of Control Variables $\mathbf{u}, \mathbf{I}_{mu}$

The updated control variables $\mathbf{u} = \mathbf{u}^o + \Delta \mathbf{u}$. The updated mismatch variables $\mathbf{I}_m = (1-v) \mathbf{I}_m^o$, specifically $I_{mukr} = (1-v) I_{mkr}^o$ and $I_{muki} = (1-v) I_{mki}^o$.

5.3.4.4 Compute State Variables (Power Flow Solution) \mathbf{x}

The state variables of the slack bus \mathbf{x}_{slack} can be calculated via substituting into the power flow equations at the slack bus. The state variables of other buses $\mathbf{x}_{reduced}$ can be solved by the Newton-Raphson method according to the reduced power flow equations shown as follows:

1. Let $w = 0$.
2. Assume an initial guess \mathbf{x}^0 for \mathbf{x} .
3. Compute $\mathbf{g}_{reduced}(\mathbf{x}^w_{reduced}, \mathbf{u})$. If $\|\mathbf{g}_{reduced}(\mathbf{x}^w_{reduced}, \mathbf{u})\| \leq \varepsilon$, $\mathbf{x}^w_{reduced}$ is the solution and the procedure is terminated. Otherwise, go to step 4.
4. Compute the Jacobian matrix: $\frac{\partial \mathbf{g}_{reduced}(\mathbf{x}^w_{reduced}, \mathbf{u})}{\partial \mathbf{x}_{reduced}}$.
5. Compute $\mathbf{x}^{w+1}_{reduced} = \mathbf{x}^w_{reduced} - \left(\frac{\partial \mathbf{g}_{reduced}(\mathbf{x}^w_{reduced}, \mathbf{u})}{\partial \mathbf{x}} \right)^{-1} \mathbf{g}_{reduced}(\mathbf{x}^w_{reduced}, \mathbf{u})$. (5-13)
6. $w = w + 1$. If $w \leq w_{max}$, go to step 2; otherwise, return nonconvergence. ($w_{max} = 20$ in this algorithm.)

5.3.4.5 New Operating Point ($\mathbf{x}, \mathbf{u}, \mathbf{I}_m$)

The values computed in the above two steps form a new operating point. Set $\Delta \mathbf{u} = \mathbf{0}$ in this step.

5.3.5 Eliminate Violations in the Modeled System

Some operating constraints, especially those reaching upper or lower bounds in the LP problem, are out of their bounds due to the linearization errors. These violations are slight since the linearization errors are limited in a small region. To eliminate these slight violations, the algorithm retrieves the previous solution and changes the \mathbf{b} vector in the LP problem.

5.3.5.1 Check Violation for the Modeled Constraints

This step checks all the constraints already included in the model. It is checked whether $\mathbf{h}^{\min} \leq \mathbf{h}(\mathbf{x}, \mathbf{u}) \leq \mathbf{h}^{\max}$ holds at the new operating point. If any modeled constraint is not satisfied, the algorithm updates its corresponding constant item in \mathbf{b} , retrieves the previous solution, and solves the updated problem. Then, the algorithm continues to check all other operating constraints.

5.3.5.2 Update LP Problem, Compute New \mathbf{b} Vector

The detailed explanation of this step is shown in Appendix B.4. Formulas are listed as follows:

The new value of b for an upper bound constraint is

$$\begin{aligned} & \Delta h^{\max}, & \text{if } h(\mathbf{x}, \mathbf{u}) \leq h^{\max} \text{ is not violated;} \\ & \Delta h^{\max} - [h(\mathbf{x}, \mathbf{u}) - h^{\max}] \\ & - \left\{ \Delta h^{\max} - \sum_{u_i \in \mathbf{u}} \left[\Delta u_i \left(\frac{\partial \mathbf{h}(\mathbf{x}^o, \mathbf{u}^o)}{\partial \mathbf{u}} \right) \Big|_{\mathbf{g}_{\text{reduced}}(\mathbf{x}, \mathbf{u}, \mathbf{v})=0} \right] - \mathbf{v} \cdot \frac{\partial \mathbf{h}(\mathbf{x}^o, \mathbf{u}^o)}{\partial \mathbf{v}} \Big|_{\mathbf{g}_{\text{reduced}}(\mathbf{x}, \mathbf{u}, \mathbf{v})=0} \right\}. \end{aligned} \quad (5-14)$$

if $h(\mathbf{x}, \mathbf{u}) \leq h^{\max}$ is violated.

The new value of b for a lower bound constraint is

$$\begin{aligned} & \Delta h^{\min}, & \text{if } h^{\min} \leq h(\mathbf{x}, \mathbf{u}) \text{ is not violated;} \\ & \Delta h^{\min} + [h^{\min} - h(\mathbf{x}, \mathbf{u})] \\ & + \left\{ \Delta h^{\min} - \sum_{u_i \in \mathbf{u}} \left[\Delta u_i \left(\frac{\partial \mathbf{h}(\mathbf{x}^o, \mathbf{u}^o)}{\partial \mathbf{u}} \right) \Big|_{\mathbf{g}_{\text{reduced}}(\mathbf{x}, \mathbf{u}, \mathbf{v})=0} \right] - \mathbf{v} \cdot \frac{\partial \mathbf{h}(\mathbf{x}^o, \mathbf{u}^o)}{\partial \mathbf{v}} \Big|_{\mathbf{g}_{\text{reduced}}(\mathbf{x}, \mathbf{u}, \mathbf{v})=0} \right\}. \end{aligned} \quad (5-15)$$

if $h^{\min} \leq h(\mathbf{x}, \mathbf{u})$ is violated.

5.3.5.3 Retrieve Operation Point ($\mathbf{x}^o, \mathbf{u}^o, \mathbf{I}_m^o$)

That is $\mathbf{x} = \mathbf{x}^o$, $\mathbf{u} = \mathbf{u}^o$, and $\mathbf{I}_m = \mathbf{I}_m^o$.

5.3.6 Procedures for Solving the Violation for All Constraints

Since the linearized problem does not include all the operating constraints, the power flow solution may not satisfy some unmodeled operating constraints. Therefore, the algorithm adds those violated constraints, retrieves the previous operating point (\mathbf{x}^o , \mathbf{u}^o , \mathbf{I}_m^o), linearizes new constraints, and solves the updated linearized optimization problem and the power flow.

5.3.6.1 Check Violation for All Constraints

This step checks all the operating constraints. If any unmodeled constraint $h^{\min} \leq h(\mathbf{x}, \mathbf{u})$ or $h(\mathbf{x}, \mathbf{u}) \leq h^{\max}$ is not satisfied, the algorithm continues this procedure; otherwise, the algorithm checks whether mismatches are all zero.

5.3.6.2 Add New Violated Constraints to the Model

If $h(\mathbf{x}, \mathbf{u})$ is below its lower bound and is not modeled, the algorithm adds $h^{\min} \leq h(\mathbf{x}, \mathbf{u})$. If $h(\mathbf{x}, \mathbf{u})$ is above its upper bound and is not modeled, the algorithm adds $h(\mathbf{x}, \mathbf{u}) \leq h^{\max}$.

5.3.6.3 Retrieve Operation Point (\mathbf{x}^o , \mathbf{u}^o , \mathbf{I}_m^o)

That is $\mathbf{x} = \mathbf{x}^o$, $\mathbf{u} = \mathbf{u}^o$, and $\mathbf{I}_m = \mathbf{I}_m^o$.

5.3.6.4 Linearize New Constraints

This step is the same as the step linearizing operating constraints in Section 5.3.3.2.

5.3.7 Procedures for the Next Iteration

5.3.7.1 Zero Mismatches?

If all mismatches (I_{mkr} , I_{mki}) are zero, the optimal solution is found. Otherwise, the algorithm stores the operating point and processes to the next iteration.

5.3.7.2 Store Operating Point ($\mathbf{x}, \mathbf{u}, \mathbf{I}_m$) to ($\mathbf{x}^o, \mathbf{u}^o, \mathbf{I}_m^o$)

That is $\mathbf{x}^o = \mathbf{x}$, $\mathbf{u}^o = \mathbf{u}$, and $\mathbf{I}_m^o = \mathbf{I}_m$.

5.3.7.3 Reclassify State and Control Variables

The power flow may fail to converge after one iteration or give a solution with very large violations at some state variables, while the system has a valid power flow solution. This usually occurs when the moving ranges of the control variables are too large. Using smaller ranges can avoid this circumstance, but will reduce the convergence speed of the algorithm. Sometimes reclassifying state and control variables is another option to solve these power flow violations.

If the reactive power output of a generator at a PV bus is far out of its bound, the algorithm will change the bus type to PQ mode. That is, Q_g becomes a control variable, and V_{mag} becomes a state variable. On the other hand, if the voltage magnitude at a PQ bus is far out of its bound, the algorithm will change the bus type to PV mode. That is, V_{mag} becomes a control variable, and Q_g becomes a state variable.

5.4 SQP Algorithm Implementation

This section presents the SQP implementation which solves OPF iteratively. QP usually can be formulated as

$$\begin{aligned} \min f(\mathbf{x}) &= 1/2 \mathbf{x}^T \mathbf{Q} \mathbf{x} + \mathbf{c}^T \mathbf{x} \\ \text{st. } \quad \mathbf{A} \mathbf{x} &\leq \mathbf{b} \\ \mathbf{E} \mathbf{x} &= \mathbf{d} \end{aligned} \tag{5-16}$$

where \mathbf{Q} is a positive semi-definite matrix. Since the cost-driven objective function of the OPF problem demonstrated in this thesis is quadratic and the coefficients of the quadratic items are positive, \mathbf{Q} is positive semi-defined. Therefore, the OPF problem can be solved via SQP which needs to linearize the constraints and keep the objective function quadratic only. SQP has a similar routine as SLP.

This dissertation work solves QP via the Gurobi solver [123]. Although QP has the same objective function as the original nonlinear optimization problem does, SQP may not perform better than SLP in solving the OPF problem, because of the following reasons: First, both SLP and SQP linearize all the operating constraints and will introduce a linearization error. If the benefit from the unchanged objective function cannot compensate the linearization error in the final step, SQP cannot lead to a better result than SLP. Second, SLP usually needs less runtime since LP solvers are faster than QP solvers. Therefore, this work selects SLP as the main implementation and use SQP for comparison.

5.5 Parallelism in OPF

Multi-core processors are a major development trend in computer science nowadays. A multi-core processor contains several central processing unit (CPU) cores. There are usually even numbers of cores in one processor due to manufacture benefits. Ideally, a dual-core processor runs twice as fast as a sole-core processor does under the same manufacturing and design technologies. However, the performance gain using a multi-core processor highly depends on the algorithm design and the software implementation. For example, a multi-core processor performs as well as a sole-core processor does on the algorithm with no parallelism. Figure 5.4 shows a quad-core processor architecture for demonstration [93]. A sequential program will always visit one of these four cores during the iterations. If the program runs on Core 1, other cores will not be visited since the lines must run in sequence.

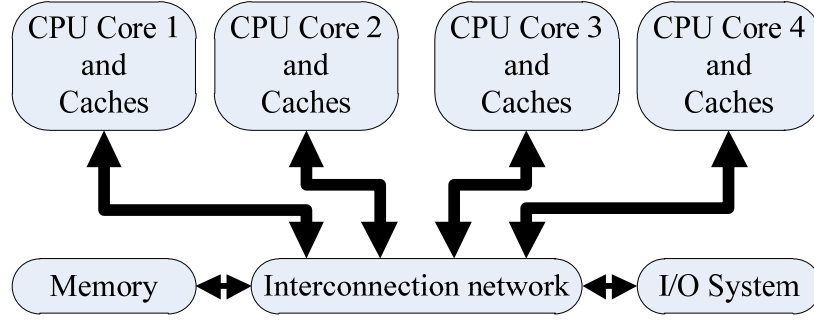


Figure 5.2 A generic quad-core processor

Parallelism requires different programming styles from traditional sequential designs. In this OPF algorithm, most of runtime is consumed in three steps: linearizing constraints, solving the LP or QP problem, and solving the power flow. This section focuses on paralleling the linearization step. LP and QP are solved via open-source or commercial optimization solvers such as GLPK [122] and Gurobi [123]. Here the Gurobi optimization solver already includes a parallel barrier solver. The parallel power flow solver is left for future development since it is not the key point in this dissertation. However, parallelism is important for all these three steps if this algorithm is released for practical or commercial use in the future.

Although operating constraints are added adaptively, the OPF algorithm may include numerous modeled constraints since the whole constraint set is very large. For example, a power system including 3,000 buses and 5,000 interconnections (transmission lines and transformers) has around 11,000 operating constraints in total if limits are applied to bus voltages and power transmissions. A three-phase unbalanced power system with the same size has around 33,000 operating constraints three times of the symmetric and balanced system. Therefore, paralleling the linearization step is essential and will result in significant performance improvement. On the other hand, linearizing one constraint will not affect the linearization of other constraints since two constraints are independent in the formulation of the optimization problem. Figure 5.3 shows the changes in the linearization step in the OPF flow chart with N operating constraints for a

computer with n CPU cores. The results in the intermediate steps, such as A_i and B_i ($i = 1, \dots, n$), should be stored separately in the memory. Otherwise, they may overlap each other resulting in incorrect linearized coefficients since parallel lines run in random order.

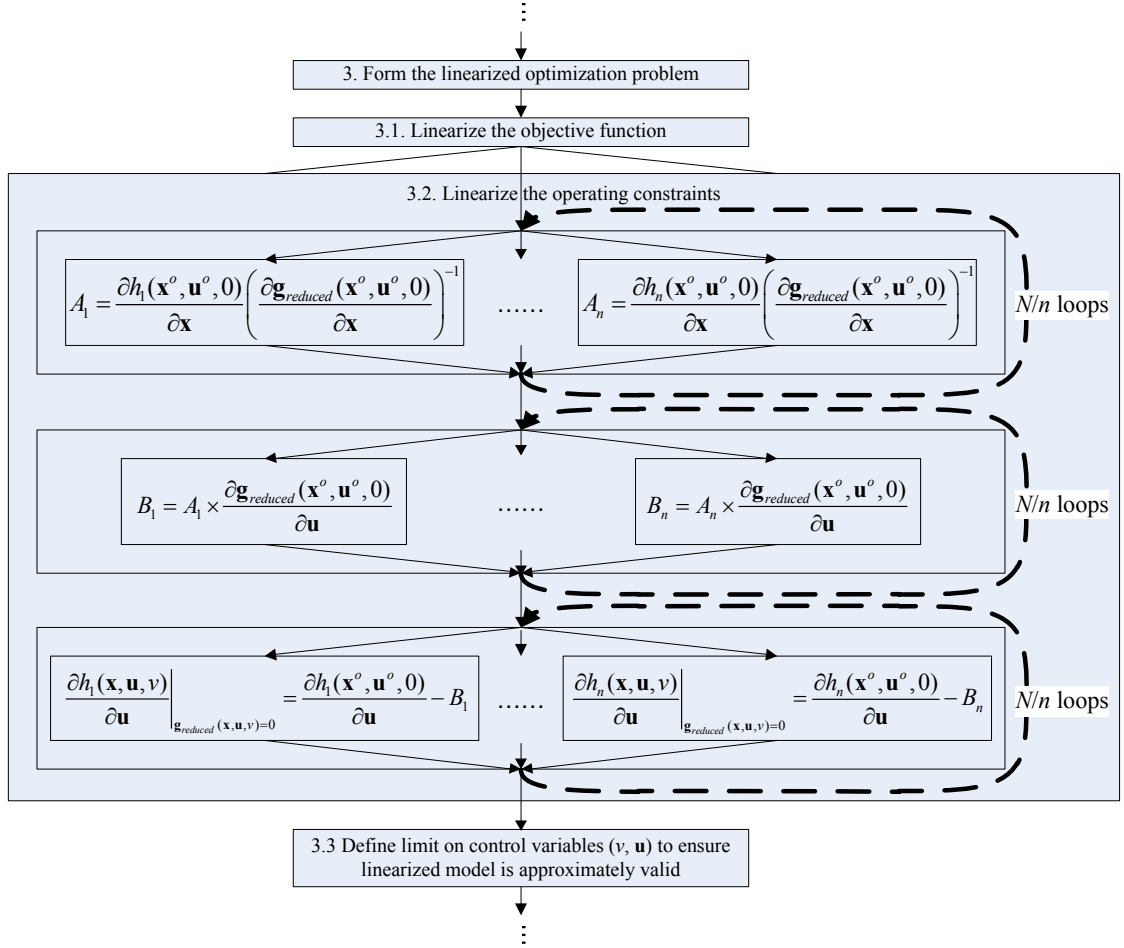


Figure 5.3 OPF flow chart update for parallelism

5.6 Post-Solution Sensitivity Analysis

This section presents the relationship between the optimal solution and the constraints. If the solution satisfies the equation condition of a constraint, it is active, otherwise the constraint is inactive. An active constraint means the system is running at its boundary and any disturbance may cause the system to collapse. An active constraint has a nonzero corresponding dual solution at the final iteration. The dual solution is

referred to as the shadow price, the change of the objective value in the optimal solution obtained by adjusting the constraint infinitesimally. The shadow price is the maximum price that the operator is willing to pay for an extra unit of given limited resource. If a constraint is inactive, its shadow price is zero. It means that changing the constraint bound does not affect the value of the cost function. If a constraint is active, its shadow price is nonzero.

The constraints of the control variable u_i are $u_i^{\min} \leq u_i \leq u_i^{\max}$, where the shadow prices are $\mu_i^{\min} = \frac{dJ}{du_i^{\min}}$ and $\mu_i^{\max} = \frac{dJ}{du_i^{\max}}$. The shadow prices tell the system planner how to make the new system more profitable by changing constraint bounds. For example, it is better to enlarge the limit of a generator with larger μ_i^{\max} . In addition, the bus with these generators is also a better location for new generators if needed.

The shadow prices of the operating constraints $h_i^{\min} \leq h_i(\mathbf{x}, \mathbf{u}) \leq h_i^{\max}$ ($\lambda_i^{\min} = \frac{dJ}{dh_i^{\min}}$ and $\lambda_i^{\max} = \frac{dJ}{dh_i^{\max}}$) have the same property as μ_i^{\min} and μ_i^{\max} , although $h_i(\mathbf{x}, \mathbf{u})$ in the power flow solution may not reach the constraint bounds due to system nonlinearity. The tighter the limit on a power transmission line, the higher the cost and the smaller the feasible region will be. A reasonable V_{kmag}^{\min} and V_{kmag}^{\max} help the system to be feasible since bus voltages far from 1.0 pu may cause the system to be unstable.

5.7 The OPF Software Design

OPF is a complicated software tool to provide the optimal status based on the present status of the system. Since power systems are very large nowadays, OPF software should be properly designed to ensure runtime efficiency. According to the symmetric and balanced power system structure and the proposed algorithms, The OPF software structure is designed as in Figure 5.4. The software has four levels: the device level, the

bus level, the system level, and the algorithm level. The device level, the bus level, and the algorithm level each have one type of data structures. They are device structures, bus structures, and an optimization structure. The system level includes two types of data structures: power flow structures and variable structures. A structure communicates with the others via functions connected between them.

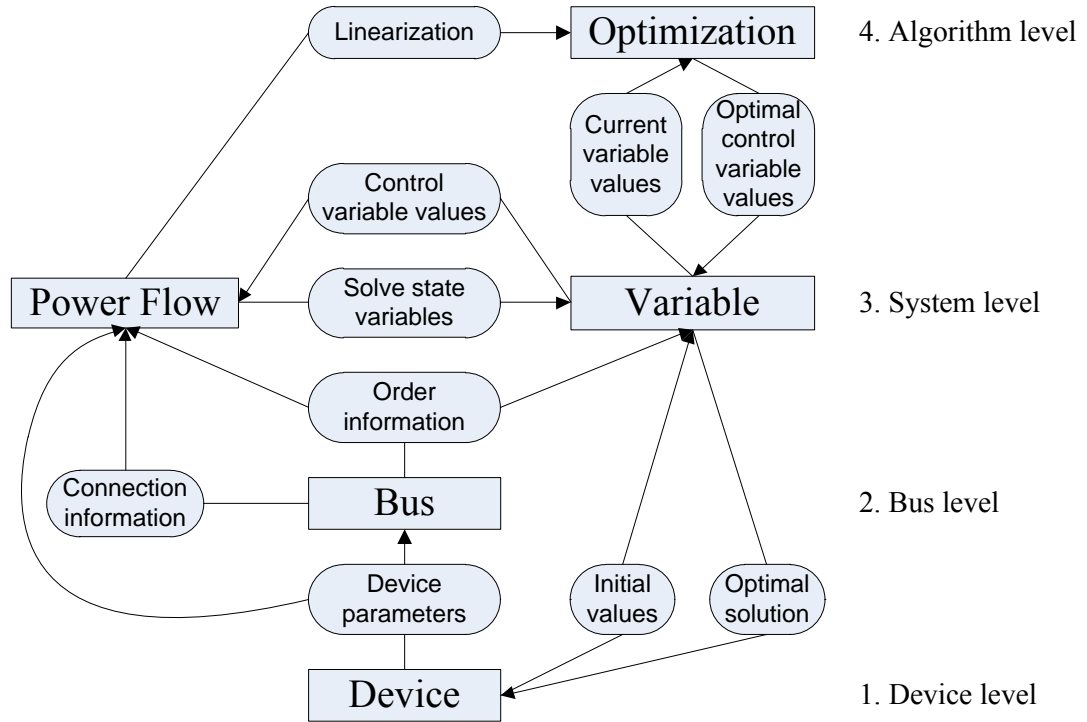


Figure 5.4 The OPF software design

Device structures store the device parameters and the state values. All the device structures are sorted according to their connected buses. The connection information between adjacent buses is provided by the transmission lines and the transformers. In the system level, the power flow structures and the variables are also sorted according to the buses. Power flows are used to compute the Jacobian matrix in solving the state variables and linearizing formulas for the optimization structure. The variable structures are the core structures in the OPF software. They read the initial values from the device structures at the beginning of the algorithm and write the optimal solution to them in the end. In each iteration, the variable structures send the present values of the control

variables from the power flow structures and obtain the solved values of the state variables from them. In the optimization step, variable structures write the present variable values to the optimization structure before LP or QP and write the updated optimal values of the control variables afterwards.

This software design fits for the single-phase OPF problem since the Jacobian matrix computed from the power flow equations can be easily ordered with large diagonal elements according to the bus order and device types. The Jacobian matrix with large diagonal elements can improve the computational efficiency of solving the power flow. The power flow equations are sorted according to the bus order. The first two power flow equations of a bus are the current balance equations and the device equations are listed after those. The order of the device equations should ensure that the abstract values of the derivatives at diagonal elements are larger than or equal to the abstract values of all other derivatives at the same row and column in the Jacobian matrix. Symmetric and balanced power systems are much simpler than three-phase power systems which require an automatic algorithm to ensure that the Jacobian matrix has large diagonal elements. Therefore, Section 6.6 presents the software design of three-phase OPF different from the design in Figure 5.4.

5.8 Summary

This chapter presented a robust and high-efficient OPF algorithm using the sequential methods to address the shortcomings of present OPF algorithms. They are classified into three categories: (a) nonlinear programming (NLP), (b) intelligent search methods, and (c) sequential algorithms. Their shortcomings are summarized as follows: First, all these algorithms require a feasible power flow solution as the initial working point and iteratively optimize the current working point to reach the optimal solution. Second, the efficiency of these algorithms needs to be improved. For example, they include all power flow equations in the constraint set, while only two system-level power

balance equations are needed in the proposed algorithm. In addition, NLP algorithms include all the operating constraints in their Karush-Kuhn-Tucker (KKT) conditions. Intelligent search methods are first-order methods and inefficient for large-scale systems since they have their own strategies which are less relevant to the system structure.

Robustness means the algorithm can provide a solution for any problem. This algorithm starts from an infeasible optimal state and moves to the feasible region while maintaining an optimal status. System feasibility is maintained by introducing artificial mismatch current sources at each bus. The mismatches reduce iteratively and the optimization method ensures that the solution is optimal at each iteration. If the feasible solution is found, it is optimal. Otherwise, the algorithm returns a suboptimal point providing the best choice to solve system infeasibility with a set of remedial actions.

High efficiency means less runtime. First, the algorithm models OPF as a quadratic problem for fast convergence in solving the power flow. Therefore, the formulated optimization problem is a quadratic optimization problem. Second, the algorithm identifies active constraints and adds them to the modeled constraint set if needed. For example, power flow equations are replaced by two current conservation equations at the system level, operating constraints are added when they are violated in the previous iteration, and the mismatch variables are represented by one control variable. Third, a sparsity technology is introduced in the matrix computation for large-scale systems.

SQP has the same routine as SLP has, while the objective function of SQP is quadratic. Although QP preserves more information than LP in the objective function of the converted problem, the performance of SQP may be worse. Then, the discussion of parallelism showed that parallel programming on multi-core or multi-CPU hardware platforms will improve the runtime. Next, this chapter analyzed the sensitivity of constraints via the small disturbance method on constraint boundaries. Finally, the software design of the proposed algorithm was described.

CHAPTER 6

THREE-PHASE OPTIMAL POWER FLOW

6.1 Introduction

A major goal of smart grid technologies is to extend transmission grid analysis and control methods to distribution systems. Hence, distribution management systems (DMS) for the smart grid need to include functions such as state estimation [91], [92] and optimal power flow [94]-[102] which are common in energy management systems (EMS). Since distribution systems generally operate in unbalanced conditions, three-phase optimal power flow (TOPF) is required rather than traditional single-phase OPF. This chapter proposes a TOPF formulation and a solution algorithm that operates in the infeasible region and moves the operating point to a feasible and optimal point via sequential methods. The proposed TOPF formulation is similar as the single-phase OPF formulation. However, they are not exactly the same. There are four complex voltage variables at each bus in three-phase unbalanced power systems. In addition, TOPF includes both continuous and discrete variables. Therefore, a TOPF algorithm is proposed based on the OPF algorithm with some modifications.

6.2 Three-Phase OPF

The cost function of TOPF is similar to that of the single-phase OPF. This cost function is the sum of the mismatch penalties of each phase and the quadratic cost functions of all the generators. The mismatch variables are the real and imaginary currents injected to all phases of each bus in the quadratic three-phase power system model. Therefore, the TOPF problem is

$$\begin{aligned}
\min J(\mathbf{x}, \mathbf{u}) &= \mu \sum (|\mathbf{I}_m|) + \sum_{k=1}^N \sum_{j=1}^{M(k)} c_{k,j}(\mathbf{x}, \mathbf{u}) \\
s.t. \quad &\mathbf{g}(\mathbf{x}, \mathbf{u}, \mathbf{I}_m) = 0 \\
&\mathbf{h}^{\min} \leq \mathbf{h}(\mathbf{x}, \mathbf{u}) \leq \mathbf{h}^{\max} \quad , \\
&\mathbf{u}_c^{\min} \leq \mathbf{u}_c \leq \mathbf{u}_c^{\max} \\
&\mathbf{u}_d = 0 \text{ or } 1
\end{aligned} \tag{6-1}$$

where

\mathbf{x} is the state variable vector,

\mathbf{u}_c is the vector of the continuous control variables,

\mathbf{u}_d is the vector of the integer control variables, ($\mathbf{u} = [\mathbf{u}_c^T, \mathbf{u}_d^T]^T$)

\mathbf{I}_m is the vector of the mismatch currents,

$J(\mathbf{x}, \mathbf{u})$ is the objective function, which takes into account the operation cost of the system and mismatch penalties,

N is the total number of the buses in the power system,

$M(k)$ is the number of the generators at Bus k ,

$c_{k,j}(\mathbf{x}, \mathbf{u})$ is the cost function of the j th generator at Bus k ,

$\mathbf{g}(\mathbf{x}, \mathbf{u}, \mathbf{I}_m) = 0$ are the three-phase quadratized power flow equations,

$\mathbf{h}^{\min} \leq \mathbf{h}(\mathbf{x}, \mathbf{u}) \leq \mathbf{h}^{\max}$ are the operating constraints,

$\mathbf{u}_c^{\min} \leq \mathbf{u}_c \leq \mathbf{u}_c^{\max}$ are the constraints of the continuous control variables, and

$\mathbf{u}_d = 0$ or 1 means the values of the integer control variables are 0 or 1.

Section 6.3.1 shows the detailed description of the state variables and the control variables. Section 6.4 shows the detailed description of the constraints.

An application of this TOPF problem is in the measurement of the cost of loss, which can be used as the operating cost during system planning. Since the system configurations of different planning scenarios for the same loading conditions are different, their operating costs are different, such as different VAR source locations. Therefore, the operating cost should be considered in system planning which is a cost-driven optimization problem with fixed planning intervals. The objective of this problem

is to find the planning trajectory with minimum cost. A typical method to solve a planning problem is dynamic programming since the problem satisfies the principle of optimality. All the costs are pre-computed and stored in a table, and then the algorithm looks up the table recursively to find the optimal trajectory. The detailed explanation is shown in Chapter 7. TOPF is the best choice to compute the operating cost since a system usually runs under its optimal conditions. Therefore, TOPF is a subroutine in the planning.

6.3 Three-Phase Quadratic General Bus Modeling

Since a distribution system is unbalanced, a three-phase model is required in the TOPF algorithm. A three-phase general bus includes (a) synchronous generators, (b) mismatch current sources, (c) three-phase constant power loads, (d) three-phase constant impedance loads, (e) induction motors, (f) capacitor banks, (g) inductors, (h) static VAR compensators, (i) transmission lines, (j) multiphase cables, (k) two-winding three-phase transformers, (l) three-winding three-phase transformers, (m) phase shifters, (n) single-phase transformers, (o) single-phase constant power loads, (p) single-phase constant impedance loads, (q) pluggable hybrid electric vehicles, etc. Figure 6.1 shows the one-line diagram of the general bus in a three-phase power system. The devices connected to the bus form bus resources while the interconnections between adjacent buses form network. Each bus has four complex current conservation equations, which represent phase A , B , C , and N respectively. Power flow equations consist of the current conservation equations of each phase and several internal equations for each device with internal state variables.

The power flow equations of Bus k (a three-phase line) are

$$\mathbf{g}_k(\mathbf{x}, \mathbf{u}, \mathbf{I}_m) = \begin{cases} \sum_k (\tilde{I}_{a,gk} + \tilde{I}_{a,dpk} + \tilde{I}_{a,dik} + \tilde{I}_{a,devk} + \tilde{I}_{a,ck} + \tilde{I}_{a,ik} + \tilde{I}_{a,svck} + \tilde{I}_{a,motork} + \tilde{I}_{a,k} + \tilde{I}_{a,mk}) = 0 \\ \sum_k (\tilde{I}_{b,gk} + \tilde{I}_{b,dpk} + \tilde{I}_{b,dik} + \tilde{I}_{b,devk} + \tilde{I}_{b,ck} + \tilde{I}_{b,ik} + \tilde{I}_{b,svck} + \tilde{I}_{b,motork} + \tilde{I}_{b,k} + \tilde{I}_{b,mk}) = 0 \\ \sum_k (\tilde{I}_{c,gk} + \tilde{I}_{c,dpk} + \tilde{I}_{c,dik} + \tilde{I}_{c,devk} + \tilde{I}_{c,ck} + \tilde{I}_{c,ik} + \tilde{I}_{c,svck} + \tilde{I}_{c,motork} + \tilde{I}_{c,k} + \tilde{I}_{c,mk}) = 0 \\ \sum_k (\tilde{I}_{n,gk} + \tilde{I}_{n,dpk} + \tilde{I}_{n,dik} + \tilde{I}_{n,devk} + \tilde{I}_{n,ck} + \tilde{I}_{n,ik} + \tilde{I}_{n,svck} + \tilde{I}_{n,motork} + \tilde{I}_{n,k} + \tilde{I}_{n,mk}) = 0 \\ \text{Internal equations of all devices connected to Bus } k \end{cases},$$

(6-2)

where a, b, c denote the three phases, and n denotes the neutral line.

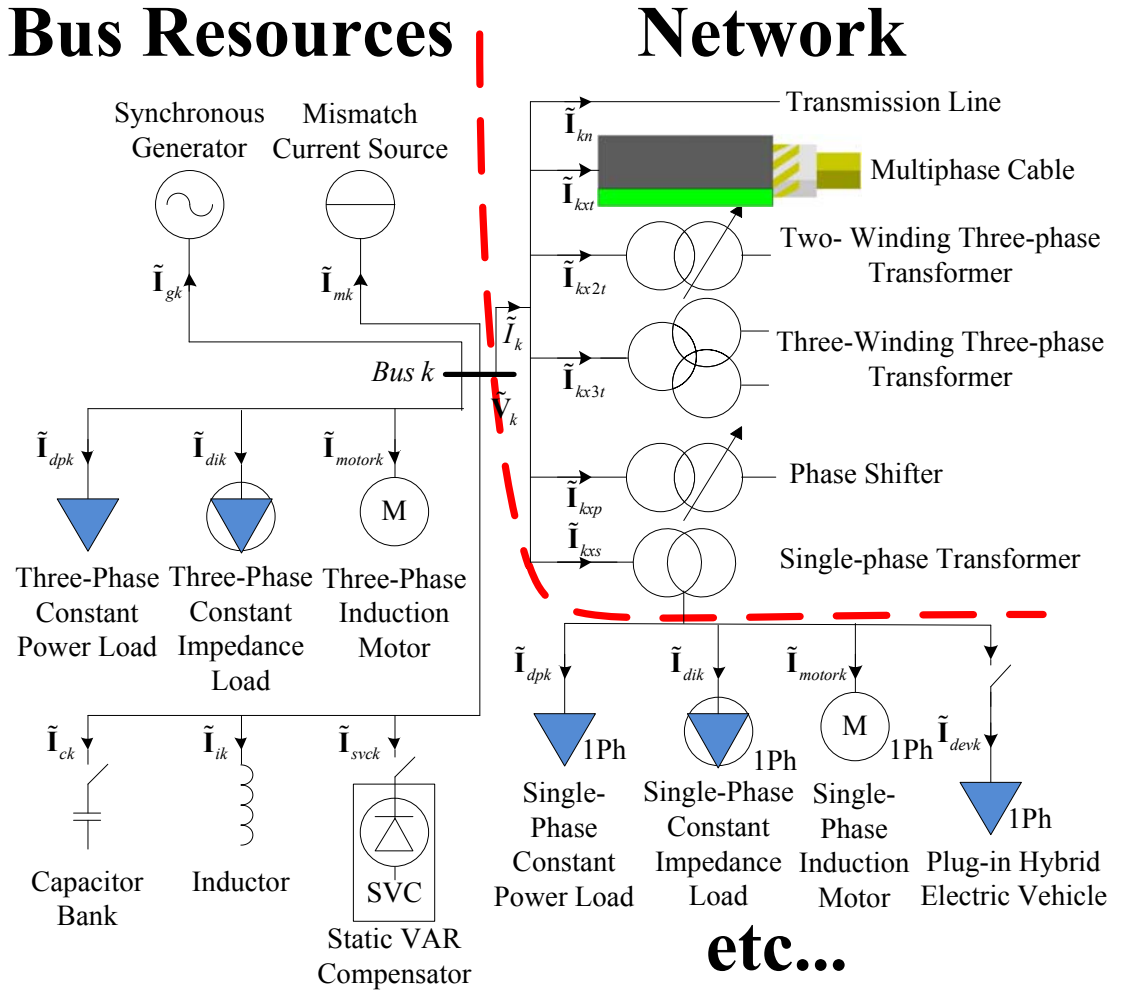


Figure 6.1 The one-line diagram of a general bus in three-phase power systems

6.3.1 The Description of OPF State and Control Variables

The variables in three-phase OPF are classified into control variables (**u**) and state variables (**x**). The control variables can be adjusted directly and independently. The state variables describe the external and internal states of each device. The following subsections show the variable classification of various devices.

6.3.1.1 Synchronous Generators

The variables of a synchronous generator are classified according to their control options for operating the generator. This classification is as follows:

- PQ mode:

$$u = [P, Q]^T,$$

$$x = [V_{ar}, V_{ai}, V_{br}, V_{bi}, V_{cr}, V_{ci}, V_{nr}, V_{ni}, E_r, E_i]^T,$$

- PV mode:

$$u = [P, V_{mag}]^T,$$

$$x = [V_{ar}, V_{ai}, V_{br}, V_{bi}, V_{cr}, V_{ci}, V_{nr}, V_{ni}, E_r, E_i]^T,$$

- Slack mode:

$$u = [V_{mag}]^T,$$

$$x = [V_{ar}, V_{ai}, V_{br}, V_{bi}, V_{cr}, V_{ci}, V_{nr}, V_{ni}, E_r, E_i]^T,$$

where

P is the real power output of the generator,

Q is the reactive power output of the generator,

V_{mag} is the line-to-line voltage magnitude,

V_{ar} and V_{ai} are real and imaginary parts of phase A voltage,

V_{br} and V_{bi} are real and imaginary parts of phase B voltage,

V_{cr} and V_{ci} are real and imaginary parts of phase C voltage,

V_{nr} and V_{ni} are real and imaginary parts of neutral line voltage,

E_r and E_i are real and imaginary parts of generator EMF.

$V_{ar}, V_{ai}, V_{br}, V_{bi}, V_{cr}, V_{ci}, V_{nr},$ and V_{ni} are the external state variables, which are also mutually owned by the devices connected to the same bus. E_r, E_i are the internal state variables, where E_i at the slack bus equals zero.

6.3.1.2 Constant Power Loads

Since constant power loads are not controllable, they have state variables only.

The state variables of a three-phase constant power load are

$$x = [V_{ar}, V_{ai}, V_{br}, V_{bi}, V_{cr}, V_{ci}, V_{nr}, V_{ni}, u_{1r}, u_{1i}, u_{2r}, u_{2i}]^T,$$

where $u_{1r}, u_{1i}, u_{2r}, u_{2i}$ are internal variables introduced to quadratize the device model.

The state variables of a single-phase constant power load are

$$x = [V_{1r}, V_{1i}, V_{2r}, V_{2i}, u_{1r}, u_{1i}]^T.$$

where

V_{1r} and V_{1i} are the real and imaginary parts of Terminal 1 voltage,

V_{2r} and V_{2i} are the real and imaginary parts of Terminal 2 voltage.

A plug-in hybrid electric vehicle (PHEV) is modeled as a single-phase constant power load with a switch. Since PHEVs are heavy loads and a power system may contain a large number of PHEVs, their charging time should be well scheduled to avoid high peak load.

6.3.1.3 Constant Impedance Loads

Since a three-phase constant impedance load is not controllable and does not have any internal states, it has only external state variables:

$$x = [V_{ar}, V_{ai}, V_{br}, V_{bi}, V_{cr}, V_{ci}, V_{nr}, V_{ni}]^T.$$

A single-phase constant impedance load also has only external state variables:

$$x = [V_{1r}, V_{1i}, V_{2r}, V_{2i}, V_{nr}, V_{ni}]^T.$$

6.3.1.4 Transmission Lines and Multiphase Cables

A transmission line is not controllable and does not have any internal variables. Its state variables are formed from the voltage variables of its two terminals. The state variable set of a transmission line between Bus 1 and Bus 2 is

$$x = [V_{a,1r}, V_{a,1i}, V_{b,1r}, V_{b,1i}, V_{c,1r}, V_{c,1i}, V_{n,1r}, V_{n,1i}, V_{a,2r}, V_{a,2i}, V_{b,2r}, V_{b,2i}, V_{c,2r}, V_{c,2i}, V_{n,2r}, V_{n,2i}]^T,$$

where 1 means primary side and 2 means secondary side.

A multiphase cable also has only external state variables formed from the phase voltages at the cable terminals. For example, a three-phase cable has the same state variable set as a transmission line does.

6.3.1.5 Transformers

Transformers can be classified into three-phase transformers and single-phase transformers. They can also be classified into regulated and non-regulated transformers. Non-regulated transformers do not have any control variables. They are listed as follows:

A non-regulated two-winding three-phase transformer:

$$x = [V_{a,1r}, V_{a,1i}, V_{b,1r}, V_{b,1i}, V_{c,1r}, V_{c,1i}, V_{n,1r}, V_{n,1i}, V_{a,2r}, V_{a,2i}, V_{b,2r}, V_{b,2i}, V_{c,2r}, V_{c,2i}, E_{a,1r}, E_{a,1i}, E_{b,1r}, E_{b,1i}, E_{c,1r}, E_{c,1i}]^T,$$

where

E_{ar} , E_{ai} , E_{br} , E_{bi} , E_{cr} , and E_{ci} are the real and imaginary parts of primary side EMFs for each phase respectively.

A non-regulated three-winding three-phase transformer:

$$x = [V_{a,1r}, V_{a,1i}, V_{b,1r}, V_{b,1i}, V_{c,1r}, V_{c,1i}, V_{n,1r}, V_{n,1i}, V_{a,2r}, V_{a,2i}, V_{b,2r}, V_{b,2i}, V_{c,2r}, V_{c,2i}, E_{a,1r}, E_{a,1i}, E_{b,1r}, E_{b,1i}, E_{c,1r}, E_{c,1i}]^T.$$

A non-regulated two-winding single-phase transformer with a secondary center tap:

$$x = [V_{a,1r}, V_{a,1i}, V_{n,1r}, V_{n,1i}, V_{l1,2r}, V_{l1,2i}, V_{nn,2r}, V_{nn,2i}, V_{l2,2r}, V_{l2,2i}, E_{1r}, E_{1i}]^T.$$

A regulated transformer has one more control variable (the tap setting t) compared with the corresponding non-regulated transformer. A phase shifter is a three-phase transformer with positive or negative phase angle difference between the primary side and the secondary side. Therefore, the phase shifter model is the same as the three-phase transformer model.

6.3.1.6 Capacitor Banks

A capacitor bank is controlled via a switch. Capacitor banks can be classified into controllable and non-controllable. A non-controllable capacitor bank has state variable only, while a controllable capacitor bank has one discrete control variable, the switch u_c . If the capacitor bank is connected to the grid, then $u_c = 1$; otherwise, $u_c = 0$. For a controllable capacitor bank,

$$u = [u_c]^T,$$

$$x = [V_{ar}, V_{ai}, V_{br}, V_{bi}, V_{cr}, V_{ci}, V_{nr}, V_{ni}]^T.$$

6.3.1.7 Static VAR Compensators

A static VAR compensator (SVC) can be modeled as a capacitor with continuous switching. Since the thyristor-controlled reactor in a SVC provides smooth control, a SVC has one continuous control variable, the switch u_{svc} ($0 \leq u_{svc} \leq 1$). Therefore,

$$u = [u_{svc}]^T,$$

$$x = [V_{ar}, V_{ai}, V_{br}, V_{bi}, V_{cr}, V_{ci}, V_{nr}, V_{ni}]^T.$$

6.3.1.8 Induction Motors

Induction motors are viewed as loads and are assumed to be uncontrollable in the TOPF problem. They have only state variables.

A three phase induction motor:

$$x = [V_{ar}, V_{ai}, V_{br}, V_{bi}, V_{cr}, V_{ci}, u_{1r}, u_{1i}, \dots, u_{14r}, u_{14i}]^T,$$

A single phase induction motor:

$$\mathbf{x} = [V_{lr}, V_{li}, V_{nr}, V_{ni}, u_{lr}, u_{li}, \dots, u_{15r}, u_{15i}]^T,$$

where u_{lr} , u_{li} , \dots , u_{15r} , and u_{15i} are the internal variables introduced to quadratize the models of the induction motors.

6.3.2 The Description of OPF Mismatch Variables

Mismatch current sources represent current mismatches at each bus. Mismatch variables equal current injections into the mismatch current sources. They will be reduced in each iteration and finally reach zero if the system has a valid power flow solution. There are eight mismatch variables at each bus:

$$\mathbf{I}_m = [I_{mr}, I_{mi}, I_{nr}, I_{ni}, I_{lr}, I_{li}, I_{15r}, I_{15i}]^T.$$

6.4 TOPF Constraint Description

The constraints in the TOPF problem ensure that the three-phase power system operates at normal steady-state conditions. They include the power flow equations ($\mathbf{g}(\mathbf{x}, \mathbf{u}, \mathbf{I}_m) = 0$), the operating constraints ($\mathbf{h}^{\min} \leq \mathbf{h}(\mathbf{x}, \mathbf{u}) \leq \mathbf{h}^{\max}$), and the control variable constraints ($\mathbf{u}^{\min} \leq \mathbf{u} \leq \mathbf{u}^{\max}$).

The state variable values are determined by the three-phase quadratized power flow equations. They consist of the current conservation equations at each bus and the internal equations of each device. The power flow equations of the whole system are combined with the equations of each bus:

$$\mathbf{g}(\mathbf{x}, \mathbf{u}, \mathbf{I}_m) = \begin{cases} \sum_k (\tilde{\mathbf{I}}_{gk} + \tilde{\mathbf{I}}_{dpk} + \tilde{\mathbf{I}}_{dik} + \tilde{\mathbf{I}}_{devk} + \tilde{\mathbf{I}}_{ck} + \tilde{\mathbf{I}}_{ik} + \tilde{\mathbf{I}}_{svck} + \tilde{\mathbf{I}}_{motork} + \tilde{\mathbf{I}}_k + \tilde{\mathbf{I}}_{mk}) = 0 \\ \text{Internal equations of all devices} \end{cases}, \quad (6-3)$$

$$\text{where } \tilde{\mathbf{I}}_k = [\tilde{I}_{ak} \quad \tilde{I}_{bk} \quad \tilde{I}_{ck} \quad \tilde{I}_{nk}]^T.$$

The operating constraints are listed as follows:

- The constraints of voltage magnitudes ($V_{a,kmag}$, $V_{b,kmag}$, $V_{c,kmag}$, $V_{l1,kmag}$, $V_{l2,kmag}$) at Bus k , such as $(V_{kmag}^{\min})^2 \leq V_{a,kr}^2 + V_{a,ki}^2 \leq (V_{kmag}^{\max})^2$,

- The constraints of real power (P) outputs of the slack generator: $P^{\min} \leq P \leq P^{\max}$,
- The constraints of reactive power (Q) outputs of the slack generator and PV mode generators: $Q^{\min} \leq Q \leq Q^{\max}$,
- The constraints of current transmission through a transmission line or a multiphase cable between adjacent buses (Bus k and Bus n) for each phase ($|I_{a,kn}|$, $|I_{b,kn}|$, $|I_{c,kn}|$, $|I_{l1,kn}|$, $|I_{l2,kn}|$), such as $I_{a,knr}^2 + I_{a,kni}^2 \leq |I_{kn}^{\max}|^2$,
- The constraints of the current transmissions through a transformer between adjacent buses (Bus k and Bus x) for each phase ($|I_{a,kx}|$, $|I_{b,kx}|$, $|I_{c,kx}|$), such as $I_{a,kxr}^2 + I_{a,kxi}^2 \leq |I_{kx}^{\max}|^2$.

The maximum current of a transmission line or a multiphase cable is determined according to the type and the size given by its specification. The maximum current of a transformer is computed as follows:

$$\text{Delta connection: } I_{kx}^{\max} = \frac{MVA \text{ rating}}{\text{Primary side line-to-line kV rating}}, \quad (6-4)$$

$$\text{Wye connection: } I_{kx}^{\max} = \frac{MVA \text{ rating}}{\sqrt{3} \times \text{Primary side line-to-line kV rating}}. \quad (6-5)$$

The control variable constraints include the upper and lower bounds of all the control variables. They are listed as follows:

- The real power outputs of PQ and PV mode generators : $P^{\min} \leq P \leq P^{\max}$,
- The reactive power outputs of PQ mode generators: $Q^{\min} \leq Q \leq Q^{\max}$,
- The voltage magnitudes of the slack bus and PV buses: $V_{mag}^{\min} \leq V_{mag} \leq V_{mag}^{\max}$,
- Transformer taps: $t^{\min} \leq t \leq t^{\max}$,
- Capacitor bank switches: $t_c = 0 \text{ or } 1$,
- SVC switches: $0 \leq t_{svc} \leq 1$.

6.5 Algorithm Description

The TOPF algorithm outline is the same as the OPF algorithm outline shown in Figure 5.1, but several detailed steps are different, such as initializing variables and solving the linearized optimization problem. This section will describe the major different steps in the TOPF algorithm.

6.5.1 Assign Initial Values of \mathbf{x} and \mathbf{u} to \mathbf{x}^o and \mathbf{u}^o

6.5.1.1 Assign Initial Values for Control Variables

The initial state of a power system should satisfy all the constraints while the mismatch variables may be nonzero. Therefore, the initial values of the control variables are set to some specific values within their physical limits.

6.5.1.2 Assign Initial Values for External State Variables

The state variables are classified into the external state variables and the internal state variables. The external state variables are the terminal voltages which should be equal to each other when the devices connected to the same bus. The terminal voltages of a device at a three-phase bus are initialized as

$$\text{Phase } A \text{ voltage: } V_a = \frac{(\cos \varphi + j \sin \varphi) V_{l-l}^{\text{rated}}}{\sqrt{3}}, \quad (6-6)$$

$$\text{Phase } B \text{ voltage: } V_b = \frac{\left(\cos(\varphi - \frac{2}{3}\pi) + j \sin(\varphi - \frac{2}{3}\pi) \right) V_{l-l}^{\text{rated}}}{\sqrt{3}}, \quad (6-7)$$

$$\text{Phase } C \text{ voltage: } V_c = \frac{\left(\cos(\varphi + \frac{2}{3}\pi) + j \sin(\varphi + \frac{2}{3}\pi) \right) V_{l-l}^{\text{rated}}}{\sqrt{3}} \text{ and,} \quad (6-8)$$

$$\text{Phase } N \text{ voltage: } V_n = V_{\text{neutral}}^{\text{initial}}, \quad (6-9)$$

where φ is the initial phase, V_{l-l}^{rated} is the rated line-to-line voltage, and $V_{neutral}^{nominal}$ is the initial neutral line voltage. ($V_{neutral}^{nominal}$ is set to $0.0001V$ in this algorithm.) For example, if $\varphi = 0$ at the primary side of a standard delta-wye connection transformer (leading phase angle), $\varphi = \pi/6$ at the secondary side.

The terminal voltages at the secondary side of a single-phase transformer with a secondary center tap are initialized as

$$\text{Phase } L1 \text{ voltage: } V_{l1} = \frac{(\cos \varphi + j \sin \varphi) V_{side2}^{rated}}{2}, \quad (6-10)$$

$$\text{Phase } L2 \text{ voltage: } V_{l2} = \frac{(\cos \varphi - j \sin \varphi) V_{side2}^{rated}}{2}, \text{ and} \quad (6-11)$$

$$\text{Phase } NN \text{ voltage: } V_{nn} = V_{neutral}^{initial}, \quad (6-12)$$

where φ equals the initial phase at the primary side of that transformer and V_{side2}^{rated} is the secondary side nominal voltage.

Terminal voltages of a device at a single-phase bus (branches from the secondary side of a single-phase transformer) are initialized as

$$\text{Phase } L1 \text{ voltage: } V_{l1} = (\cos \varphi + j \sin \varphi) V_{device}^{rated} \text{ and} \quad (6-13)$$

$$\text{Phase } NN \text{ voltage: } V_{nn} = V_{neutral}^{initial}, \quad (6-14)$$

where V_{device}^{rated} is the device nominal voltage.

6.5.1.3 Assign Initial Values for Internal State Variables

After the initial values of the external state variables are obtained, the algorithm will assign the initial values of the internal state variables satisfying all the internal power flow equations. This section proposes a general assignment method using Newton's method. The reasons why the algorithm does not use some more direct methods are listed as follows: First, the internal state variables cannot be assigned to some standard values like external state variables since the internal power flow equations do not include

mismatch variables. Second, the internal state variables cannot be computed via substituting formulas as was done in Section 5.3.1.2 since three-phase power systems include various types of devices and some device models have tens of state variables and power flow equations. For example, the three-phase inductor model has 28 internal state variables and the single-phase inductor model has 30 internal state variables. Therefore, the substitution method will have to develop formulas for all these state variables and some of them are very complicated.

The initial values of the internal state variables of Device k can be solved via Newton's method using a general internal device model as follows:

$$0 = Y_{eq_real_internal}^k Y^k + \begin{bmatrix} x^{kT} f_{eq_real1}^k x^k \\ x^{kT} f_{eq_real2}^k x^k \\ \vdots \end{bmatrix} - b_{eq_real_internal}^k, \quad (6-15)$$

where

Y^k is the internal state variable vector,

X^k is the state variable vector, and

$Y_{eq_real_internal}^k$ and $b_{eq_real_internal}^k$ represent the linear and constant items of the internal power flow equations respectively. This model is generated by removing the external power flow equations from the full quadratic model of Device k :

$$\begin{bmatrix} I^k \\ 0 \end{bmatrix} = Y_{eq_real}^k \begin{bmatrix} V^k \\ Y^k \end{bmatrix} + \begin{bmatrix} x^{kT} f_{eq_real1}^k x^k \\ x^{kT} f_{eq_real2}^k x^k \\ \vdots \end{bmatrix} - b_{eq_real}^k, \quad (6-16)$$

where

I^k is the through variable vector,

V^k is the external state variable vector ($X^k = [V^{kT} Y^{kT}]^T$), and

$f_{eq_real}^k$, $Y_{eq_real}^k$, and $b_{eq_real}^k$ represent quadratic, linear, and constant items respectively.

Since the external power flow equations are all linear, the quadratic items in Equation

(6-15) and Equation (6-16) are the same. The initial guess of X^* in Newton's method can be assigned to some standard values. For example, the initial guess of the electromotive force (EMF) of a three-phase transformer equals to its terminal voltages: $\tilde{E}_a = \tilde{V}_a$, $\tilde{E}_b = \tilde{V}_b$, and $\tilde{E}_c = \tilde{V}_c$. Newton's method converges within two iterations since the internal device model is either linear or quadratic.

This assignment method is effective for most of device models since their internal device models are solvable and contain all the internal state variables. However, slack and PV synchronous generators are the exceptions. For example, the internal model of a slack mode generator is

$$\begin{aligned} 0 &= E_i \\ 0 &= V_{ar}^2 + V_{ai}^2 + V_{br}^2 + V_{bi}^2 - 2V_{ar}V_{br} - 2V_{ai}V_{bi} - V_{mag}^2, \end{aligned} \quad (6-17)$$

where E_i is the imaginary part of the generator EMF and V_{mag} is the line-to-line voltage magnitude. Obviously, the real part of the generator EMF (E_r) cannot be solved from Equation (6-17). Therefore, the algorithm solves E_r and E_i in slack and PV generators using the real and reactive power equations:

$$\begin{aligned} 0 &= V_{ar}I_{ar} + V_{ai}I_{ai} + V_{br}I_{br} + V_{bi}I_{bi} + V_{cr}I_{cr} + V_{ci}I_{ci} + P \\ 0 &= -V_{ar}I_{ai} + V_{ai}I_{ar} - V_{br}I_{bi} + V_{bi}I_{br} - V_{cr}I_{ci} + V_{ci}I_{cr} + Q, \end{aligned} \quad (6-18)$$

where

$$\begin{aligned} I_{ar} &= bV_{ar} + gV_{ai} - gV_{ni} - bV_{nr} - gE_i - bE_r, \\ I_{ai} &= gV_{ar} - bV_{ai} - gV_{nr} + bV_{ni} - gE_r + bE_i, \\ I_{br} &= bV_{br} + gV_{bi} - bV_{nr} - gV_{ni} + \left(\frac{\sqrt{3}}{2}g + \frac{1}{2}b\right)E_r + \left(\frac{1}{2}g - \frac{\sqrt{3}}{2}b\right)E_i, \\ I_{bi} &= gV_{br} - bV_{bi} - gV_{nr} + bV_{ni} + \left(\frac{1}{2}g - \frac{\sqrt{3}}{2}b\right)E_r + \left(-\frac{\sqrt{3}}{2}g - \frac{1}{2}b\right)E_i, \end{aligned}$$

$$I_{cr} = bV_{cr} + gV_{ci} - bV_{nr} - gV_{ni} + \left(\frac{1}{2}b - \frac{\sqrt{3}}{2}g\right)E_r + \left(\frac{1}{2}g + \frac{\sqrt{3}}{2}b\right)E_i, \text{ and}$$

$$I_{ci} = gV_{cr} - bV_{ci} - gV_{nr} + bV_{ni} + \left(\frac{1}{2}g + \frac{\sqrt{3}}{2}b\right)E_r + \left(\frac{\sqrt{3}}{2}g - \frac{1}{2}b\right)E_i,$$

where $g + jb$ is machine admittance.

6.5.2 Solve the Linearized Optimization Problem ν , Δu , $\Delta \mathbf{I}_m$

Since the control variable of a capacitor bank is an integer variable, TOPF is a mixed-integer nonlinear programming problem (MINLP). According to the theory of computational complexity, MINLP is a NP-complete problem without any polynomial time algorithms. Therefore, there is no general time-efficient algorithm theoretically. The algorithm to the TOPF problem should be designed according to its characteristics shown as follows.

The TOPF problem in this chapter is a cost-driven MINLP and the objective function does not contain any integer variables, so the algorithm structure is the same as single-phase OPF with some minus changes including relaxing integer constraints to continuous conditions and rounding LP solutions of integer variables to their closer integer number. The reasons for these changes are listed as follows. First, our present algorithm framework is very time efficient. Second, the mismatch variables need several iterations to reach zero, so the iterations afterwards can reduce the effect of this relaxation. Therefore, we add an additional step after obtaining the LP solution: if $0 \leq t_c \leq 0.5$, then $t_c = 0$; otherwise, $t_c = 1$. This method may need some additional iterations and the solution may be suboptimal.

6.5.3 The Branch and Bound Algorithm to TOPF

Another typical method to solve MINLP is the branch and bound algorithm. This section presents the branch and bound algorithm to TOPF for comparison. The original

TOPF problem can be viewed as selecting the optimal configuration of the discrete variables from all possible combinations. The cost of each configuration is obtained via solving the associated continuous TOPF problem with determined \mathbf{u}_d values. The original TOPF problem (6-1) is relisted as follows:

$$\begin{aligned}
\min J(\mathbf{x}, \mathbf{u}) &= \mu \sum (|\mathbf{I}_m|) + \sum_{k=1}^N \sum_{j=1}^{M(k)} c_{k,j}(\mathbf{x}, \mathbf{u}) \\
s.t. \quad &\mathbf{g}(\mathbf{x}, \mathbf{u}, \mathbf{I}_m) = 0 \\
&\mathbf{h}^{\min} \leq \mathbf{h}(\mathbf{x}, \mathbf{u}) \leq \mathbf{h}^{\max} \\
&\mathbf{u}_c^{\min} \leq \mathbf{u}_c \leq \mathbf{u}_c^{\max} \\
&\mathbf{u}_d = 0 \text{ or } 1
\end{aligned} \tag{6-19}$$

The branch and bound algorithm to this TOPF problem is shown as follows:

1. Set the problem tree T_p to empty.
2. Relax all the integer constraints to continuous constraints, that is $0 \leq \mathbf{u}_d \leq 1$. Add the relaxed problem to T_p as the root and set it as the current subproblem SP_c .
3. Solve SP_c to obtain the objective function value J and values of \mathbf{u}_c and \mathbf{u}_d .
4. Set the lower bound of the problem $J_{Lower} = \min$ (objective function values of all subproblems in the problem set) and
Set the upper bound of the problem $J_{Upper} = \min$ (objective function values of all solved subproblems (all variables in \mathbf{u}_d are integer) in the problem set).
5. If SP_c does not have a feasible solution or $J \geq J_{Upper}$, remove SP_c from T_p . Otherwise, if all variables in \mathbf{u}_d are integer, SP_c is marked as solved.
6. If SP_c is removed or solved, go to step 8; otherwise, go to step 7.
7. If \mathbf{u}_d does not include any variable with a non-integer value, go to step 8; otherwise, select a non-integer variable u_{di} from \mathbf{u}_d , add two subproblems to T_p , one with additional constraint $u_{di} = 0$ and the other with $u_{di} = 1$ respectively.
8. If T_p has unsolved subproblems, Set SP_c to the next unsolved subproblem in T_p and go to step 3.

9. If T_p contains solved subproblems, select the solution of the solved subproblem with the minimum objective function as the final solution of TOPF. Otherwise, TOPF has no solution.

The branch and bound algorithm can obtain the optimal solution theoretically. However, it may need to compute all discrete variable combinations in the worst case. Therefore, the complexity of the branch and bound method is up to $O(2^n)$. That means the number of the continuous TOPF problems is 2^n in the worst case, where n is the number of the discrete variables. For example, the maximum number of continuous TOPF subproblems is 1024 if \mathbf{u}_d has only 10 elements.

6.6 The TOPF Software Design

The algorithm architecture of TOPF is the same as the single-phase version shown in Figure 5.1. However, a three-phase unbalanced power system is much more complicated than a symmetric and balanced power system. Therefore, TOPF software has a different structure with OPF software. Figure 6.2 shows the TOPF software structure fitting the property of three-phase power systems.

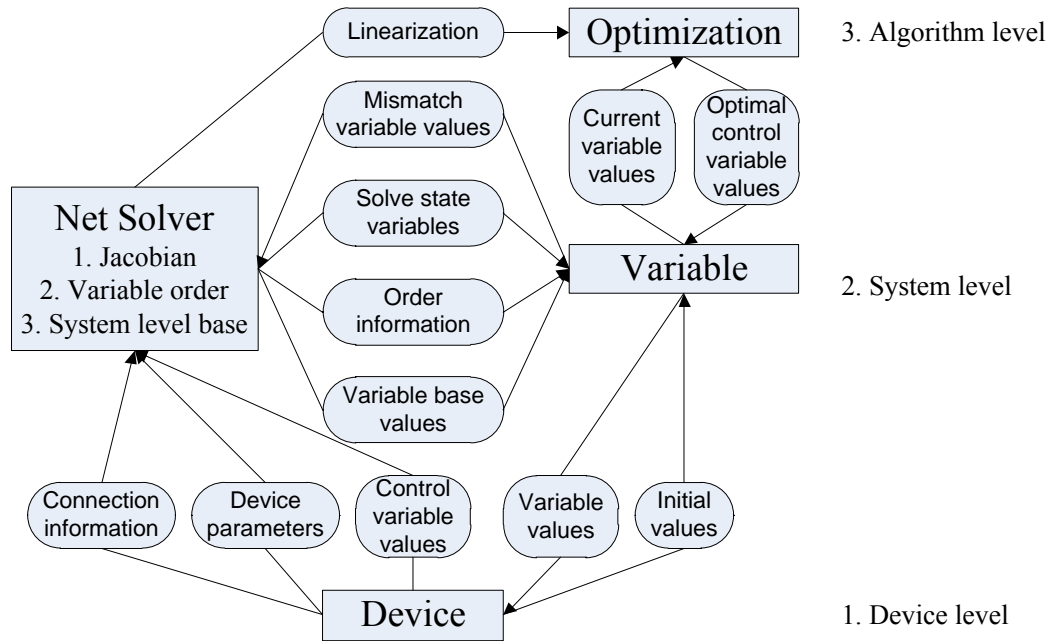


Figure 6.2 The TOPF software design

TOPF software has three levels: the device level, the system level, and the algorithm level. Since the variables are sorted according to the device types, the bus level is not needed. There are four types of data structures in these levels: device structures, a net-solver structure, variable structures, and an optimization structure. The variable structures store the system status in each iteration. They provide present the variable values to the optimization structure and receive the optimal values of the control variables after the optimization step. Figure 6.1 shows the device types included in TOPF. Although the device types are various, the data communications of the device structures are the same. The device structures send the initial variable values to the variable structures at the beginning of the algorithm and receive the update variable values from them in each iteration. Device structures also send the connection information, the device parameters, and the control variable values to Net Solver. It is the core structure providing the functions as ordering and solving the state variables, forming the Jacobian matrix in the system level, and generating the system-level bases for all variables. Net Solver also provides linearization information to the optimization structure and sends the solved state variable values, the variable orders, and the variable base values to variable structures. The detailed descriptions of Net Solver are listed as follows:

1. Net Solver orders the state variables according to both the variable types and the bus indices at the system level. The order of the state variables should ensure that the Jacobian matrix has large diagonal elements.
2. Each phase line at a bus has three base units: a voltage base, a current base, and a power base. Net Solver computes these base units using the average base units of all devices connected to that line since the base values of these devices may not be equal.
3. Since the mismatch current sources in this TOPF algorithm are not zero before the optimal solution is obtained, Net Solver solves the state variables with mismatches using the equation system:

$$0 = Y_{eq_real} \begin{bmatrix} V \\ Y \end{bmatrix} + \begin{bmatrix} x^T f_{eq_real1} x \\ x^T f_{eq_real2} x \\ \vdots \end{bmatrix} - b_{eq_real} + \begin{bmatrix} I_m \\ 0 \end{bmatrix}, \quad (6-20)$$

where

$$0 = Y_{eq_real} \begin{bmatrix} V \\ Y \end{bmatrix} + \begin{bmatrix} x^T f_{eq_real1} x \\ x^T f_{eq_real2} x \\ \vdots \end{bmatrix} - b_{eq_real} \text{ is the original quadratic power}$$

system model,

I_m is the vector of the mismatch variables, ($\# I_m = \# V$.)

Y is the vector of the internal state variables,

X is the vector of the state variables,

V is the vector of the external state variables ($X = [V^T Y^T]^T$), and

f_{eq_real} , Y_{eq_real} , and b_{eq_real} represent quadratic, linear, and constant items respectively.

6.7 Summary

This chapter first described the TOPF formulation including both continuous and discrete variables. Since TOPF includes discrete variables, an additional step is added after obtaining the updated control variable values in LP. Then, a quadratized model of three-phase power systems was presented followed by variable classification. TOPF variables are also classified into the state variables and the control variables. The state variables consist of the external state variables (terminal voltages) and the internal state variables, some of which are introduced to quadratize device models. And then, this chapter elaborated on the TOPF algorithm mainly focusing on the modifications compared with the proposed OPF algorithm. The modifications are in assigning initial variable values and in solving the linearized optimization problem. The TOPF software design was described at the end of this chapter.

CHAPTER 7

A TOPF APPLICATION – OPTIMAL VAR ALLOCATION WITH DYNAMIC PROGRAMMING

7.1 Introduction

This chapter presents the formulation and a solution method of the optimal VAR allocation problem, where TOPF is used to evaluate the performance and the cost of each decision. In general, dynamic VAR sources can mitigate fault induced delayed voltage recovery (FIDVR) phenomena, but their cost is very high. Therefore, they should be strategically placed taking into consideration both the reduction of voltage disturbance and the minimization of the total cost. Static VAR sources can help to some extent, but they cannot be switched fast enough to provide the required response. In this thesis, we use both static VAR sources (capacitor banks) and dynamic VAR sources (static VAR compensators) to improve the performance of the system under FIDVR conditions. A number of candidate locations may be selected for placing static and dynamic VAR sources. Therefore, this is a decision problem and we solve it using the dynamic programming algorithm. This decision problem has several stages and associated costs at each decision stage that can be categorized on a) economic costs and b) performance penalties. The economic costs include the annualized equivalent cost of the added equipment and the operating cost. The annualized equivalent cost is computed from the acquisition cost and the installation cost. The acquisition cost depends on equipment prices. The installation cost depends on labor prices and installation time. The operating cost cannot be easily evaluated since the actual operating status is unknown. Since a system usually runs at its optimal conditions, a good choice is to use the optimal cost under a typical operating condition, which can be computed using TOPF. The reason for

the selection of TOPF is its high computational efficiency and ability to provide accurate optimal costs under different system structures in different decision stages. The performance penalties include the penalty items of the voltage deviation at the steady state, voltage recovery time and the voltage oscillation magnitude after a fault clearing. This chapter provides the description of the planning algorithm. Chapter 10 shows the details of the planning algorithm for a power system with two distribution networks and gives the computational details of a specific state (State 6 at Stage 19) for demonstration.

7.2 Cost Definitions

Cost components are categorized on economic costs ($J_{EconomicCost}$), performance penalties ($J_{PerformancePenalty}$), and hard constraints ($J_{HardConstraint}$). Economic costs are defined as the monetary value, which are required to upgrade and operate the system. Economic costs consist of the annualized equivalent cost ($J_{AnnualizedEquivalentCost}$) and the operating cost ($J_{OperatingCost}$). Performance penalties include the voltage deviation penalty ($J_{VoltageDeviationPenalty}$), the voltage recovery time penalty ($J_{VoltageRecoveryTimePenalty}$), and the voltage oscillation penalty ($J_{VoltageOscillationPenalty}$). They are converted from their corresponding performance criteria. In addition, they occur in every planning stage recurrently, but they are different for each stage since system parameters may change during stages. Hard constraints include the voltage lower bound ($J_{VoltageLowerBound}$) and the voltage recovery time upper bound ($J_{VoltageRecoveryTimeUpperBound}$). A very high cost is assigned when a hard constraint is violated. Figure 7.1 lists the costs discussed in this section.

In summary, the total cost of a system state is

$$\begin{aligned}
 C = & J_{AnnualizedEquivalentCost} + J_{OperatingCost} \\
 & + J_{VoltageDeviationPenalty} + J_{VoltageRecoveryTimePenalty} + J_{VoltageOscillationPenalty} \\
 & + J_{VoltageLowerBound} + J_{VoltageRecoveryTimeUpperBound}
 \end{aligned} \tag{7-1}$$

The detailed definition and the computation of these costs are shown in the following subsections.

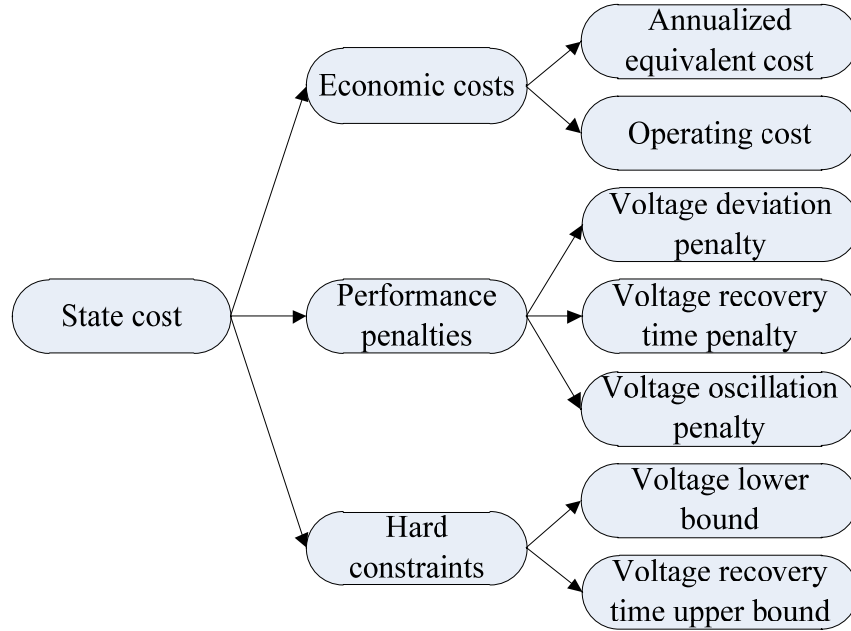


Figure 7.1 Cost classification

7.2.1 The Annualized Equivalent Cost

The annualized equivalent cost (AEC) [128] is the cost per year of owning and operating an asset over an infinite long time period that may involve replacing of equipment at the end of their economic life. AEC is computed as follows,

$$J_{\text{AnnualizedEquivalentCost}} = (A_C + I_C) \left(r + \frac{r}{(1+r)^{m/p} - 1} \right), \quad (7-2)$$

where

A_C = the acquisition cost,

I_C = the installation cost,

r = the interest rate,

m = the expected economic life time of the asset in years, and

p = the length of planning stage in years.

7.2.2 The Operating Cost

The operating cost ($J_{OperatingCost}$) is used to measure the cost of losses via TOPF for the same loading conditions with different VAR source arrangements, which may affect the operating cost due to different system configurations. TOPF evaluates the operating cost at each state and for each stage of the planning horizon. The operating cost is usually represented as a nonlinear function of the real powers generated as follows,

$$J_{OperatingCost} = \sum_{k=1}^N \sum_{j=1}^{M(k)} (a_{k,j} + b_{k,j} P_{k,j} + c_{k,j} P_{k,j}^2), \quad (7-3)$$

where

N is # the buses in the power system,

$M(k)$ is # the generators at Bus k ,

$P_{k,j}$ is the real power generated of the j th generator at Bus k ,

$a_{k,j}$, $b_{k,j}$, and $c_{k,j}$ are coefficients in the cost function of the j th generator at Bus k .

TOPF is the best choice to compute the operating cost since a three-phase power system usually operates at its optimal conditions, for which TOPF gives the minimum value of the nonlinear objective function under the operating constraints. TOPF is formed as follows:

$$\begin{aligned} J_{OperatingCost} = \min J(\mathbf{x}, \mathbf{u}) &= \mu \sum (|\mathbf{I}_m|) + \sum_{k=1}^N \sum_{j=1}^{M(k)} (a_{k,j} + b_{k,j} P_{k,j} + c_{k,j} P_{k,j}^2) \\ s.t. \quad &\mathbf{g}(\mathbf{x}, \mathbf{u}, \mathbf{I}_m) = 0 \\ &\mathbf{h}^{\min} \leq \mathbf{h}(\mathbf{x}, \mathbf{u}) \leq \mathbf{h}^{\max}, \\ &\mathbf{u}_c^{\min} \leq \mathbf{u}_c \leq \mathbf{u}_c^{\max} \\ &\mathbf{u}_d = 0 \text{ or } 1 \end{aligned} \quad (7-4)$$

where \mathbf{x} is the state variable vector, $\mathbf{u} = [\mathbf{u}_c^T, \mathbf{u}_d^T]^T$, \mathbf{u}_c is the vector of continuous control variables, \mathbf{u}_d is the vector of integer control variables, and \mathbf{I}_m is the vector of mismatch currents. $J(\mathbf{x}, \mathbf{u})$ is the objective function, which takes into account the operating cost and the mismatch penalties. When the optimal solution is found, mismatch penalties are zero

and $J(\mathbf{x}, \mathbf{u})$ equals the optimal operating cost. Chapter 6 describes the TOPF problem in detail.

7.2.3 The Voltage Deviation Penalty

The voltage deviation penalty is a penalty associated to a voltage deviation from the nominal value. The penalty is evaluated as follows: first we compute an index that quantifies the voltage deviation. Then the index is multiplied with a conversion factor that converts the voltage index into penalty. The following formula for the evaluation of the voltage deviation penalty is proposed:

$$J_{VoltageDeviationPenalty} = \beta_1 \sum_{i=1}^{N_b} S_i \left(\frac{V_{ti} - V_{ni}}{0.05V_{ni}} \right)^2, \quad (7-5)$$

where

S_i is load rating at Bus i (MW),

N_b is # the buses,

V_{ti} is the voltage magnitude of Bus i under normal operating conditions (V),

V_{ni} is the rated voltage of Bus i (V),

$\left(\frac{V_{ti} - V_{ni}}{0.05V_{ni}} \right)^2$ is the voltage deviation index, and

β_1 is the conversion factor of the voltage deviation index into penalty (\$/MW). We propose the value $\beta_1 = 2.0$ \$/MW.

7.2.4 The Voltage Recovery Time Penalty

The voltage recovery time is the time during which bus voltage remains below 90% of its nominal value after a fault clearing. The voltage recovery time penalty is evaluated as follows: first we compute an index that quantifies voltage recovery time. Then the index is multiplied with a conversion factor that converts the time index into

penalty. The following formula for the evaluation of the voltage recovery time penalty is proposed:

$$J_{VoltageRecoveryTimePenalty} = \begin{cases} 0 & , \quad \text{if } t_{ri} < 0.5s \\ \beta_2 \sum_{i=1}^{N_b} S_i \left(\frac{t_{ri} - 0.5s}{0.5s} \right)^2 & , \quad \text{if } t_{ri} \geq 0.5s \end{cases} \quad (7-6)$$

where

S_i is load rating at Bus i (MW),

N_b is # the buses,

t_{ri} is voltage recovery time of Bus i (s),

$\left(\frac{t_{ri} - 0.5s}{0.5s} \right)^2$ is the voltage recovery time index, and

β_2 is the conversion factor of the voltage recovery time index into penalty. We propose the value $\beta_2 = 30.0 \text{ \$/MW}$.

7.2.5 The Voltage Oscillation Penalty

The voltage oscillation penalty is a penalty associated with the voltage oscillation from the average value after a fault clearing. The penalty is defined as follows: first we compute an index that quantifies voltage oscillation. Then the index is multiplied with a conversion factor that converts the voltage index into penalty. The following formula for the evaluation of the voltage deviation penalty is proposed:

$$J_{VoltageOscillationCost} = \begin{cases} 0 & , \quad \text{if } V_{osci} < 0.02V_{ni} \\ \beta_3 \sum_{i=1}^{N_b} S_i \left(\frac{V_{osci} - 0.02V_{ni}}{0.02V_{ni}} \right)^2 & , \quad \text{if } V_{osci} \geq 0.02V_{ni} \end{cases} \quad (7-7)$$

where

S_i is load rating at Bus i (MW),

V_{osci} is the voltage oscillation magnitude of Bus i voltage (V),

V_{ni} is the rated voltage of Bus i (V),

N_b is # the buses,

$\left(\frac{V_{osci} - 0.02V_{ni}}{0.02V_{ni}} \right)^2$ is the voltage oscillation index, and

β_3 is the conversion factor of the voltage oscillation index into penalty. We propose the value $\beta_3 = 1.0$ (\$/MW).

7.2.6 Hard Constraint 1 – The Voltage Lower Bound

The voltage at each load bus is not allowed to be lower than 0.9 pu at the steady state. This constraint is represented as:

$$J_{VoltageLowerBound} = \begin{cases} 0, & \text{if } V_{ti} \geq 0.9V_{ni} (1, \dots, N_b) \\ \infty, & \text{if } V_{ti} < 0.9V_{ni} (1, \dots, N_b) \end{cases} \quad (7-8)$$

where

N_b is # buses,

V_{ti} is the voltage magnitude of Bus i under normal operating conditions (V), and

V_{ni} is the rated voltage of Bus i (V).

7.2.7 Hard Constraint 2 – The Voltage Recovery Time Upper Bound

The voltage recovery time at each bus is not allowed to be larger than 2 seconds after fault clearance. This constraint is represented as:

$$J_{VoltageRecoveryTimeUpperBound} = \begin{cases} 0, & \text{if } t_{ri} \leq 2s (1, \dots, N_b) \\ \infty, & \text{if } t_{ri} > 2s (1, \dots, N_b) \end{cases} \quad (7-9)$$

where

N_b is # buses and

t_{ri} is voltage recovery time of Bus i (seconds).

7.3 Candidate Location Selection

A state is defined as a specific VAR source planning configuration. For example, the configuration with no additional VAR sources is State 0. This problem has two types

of VAR sources: capacitor banks and static VAR compensators (SVC). The number of all the states is $N_c^{L_c} N_s^{L_s}$,

where

$N_c = \#$ the standard capacitor banks,

$L_c = \#$ the locations for capacitor bank installation,

$N_s = \#$ the standard SVCs, and

$L_s = \#$ the locations for SVC installation,

For example, if $N_c = 5$, $N_s = 3$, $L_c = 12$, and $L_s = 12$, the number of all the states is $1.3 \times 10^{14} = 5^{12} \times 3^{12}$. The algorithm selects the most cost efficient states for the planning problem to limit the size of the problem.

The candidate locations for additional VAR source are selected by the algorithm using sensitivity analysis. A voltage performance index is first defined and then the sensitivity of this index with respect to additions of VAR sources is computed:

$$\frac{dJ_{\text{VoltageDeviationPenalty}}}{db} = \frac{d \left(\frac{V_{ti} - V_{ni}}{0.05V_{ni}} \right)^2}{db}, \quad (7-10)$$

where

V_{ti} is the voltage magnitude of Bus i under normal operating conditions (V),

V_{ni} is the rated voltage of Bus i (V), and

b is the susceptance of the capacitor bank.

A negative sensitivity indicates that the addition of VAR sources at the specific bus will improve the voltage profile. The top two locations will be selected as candidate locations. The algorithm to select the candidate locations is designed as follows,

1. Select candidate locations for capacitor banks via sensitivity analysis.

1.1. Place a capacitor bank at Bus i .

1.2. Simulate the system and record the voltage magnitude at Bus i .

- 1.3. Compute the voltage performance index $\frac{V_{ti} - V_{ni}}{0.05V_{ni}}$ at Bus i .
- 1.4. Repeat 1.1 to 1.3 for several different sizes of capacitor banks.
- 1.5. Compute the sensitivity of performance index $\frac{d\left(\frac{V_{ti} - V_{ni}}{0.05V_{ni}}\right)^2}{db}$ at Bus i .
- 1.6. Repeat 1.1 to 1.5 for all buses.
- 1.7. Select buses with high negative sensitivity as candidate locations.
- 1.8. If two candidate locations are closer than an “electrical distance”, one is removed from the list.
2. Select candidate locations for SVCs via sensitivity analysis (procedures are the same as capacitor banks).

7.4 The Formation of States

The list of states is formed from all possible combinations of locations and resources. The number of states is $N_{c-selected}^{L_{c-selected}} N_{S-selected}^{L_{S-selected}}$, where

$N_{c-selected}$ = # selected standard capacitor banks,

$L_{c-selected}$ = # selected locations for capacitor bank installation,

$N_{S-selected}$ = # selected standard SVCs, and

$L_{S-selected}$ = # selected locations for SVC installation,

7.5 The Optimization Problem Definition

A decision problem usually has several decision stages defined by the time horizon that a decision should be made. A stage usually consists of several states. Figure 7.2 shows the dynamic programming formulation of a multistage decision problem. $X_{i,k}$ represents State i at Stage k . Using this terminology, the decision process works on the matrix of all the possible states at each stage. Specifically, the matrix shows all states in a

stage in a vertical arrangement. Decisions taken at State j at Stage $k-1$ will result in a specific state at Stage k . (Additional VAR sources may be installed.) Future decision process after Stage k depends on only the states at Stage k and is not affected by the path from the starting stage to the states at Stage k .

The reactive source planning problem is a decision problem from the initial state to the final stage. For example, the decision problem in Figure 7.2 considers a 10-year planning horizon, and each stage is assumed to be six months. The total number of stages is 20 and there are 16 states defined in each stage.

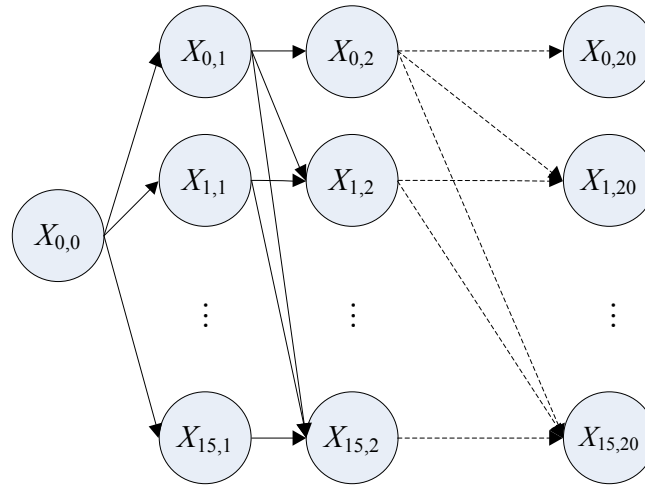


Figure 7.2 Decision tree: at each stage, there are 16 states

The objective of this decision problem is to minimize the total cost through all the planning stages. The planning problem should consider several costs, such as the acquisition cost, the installation cost, the operating cost, the voltage deviation penalty, the voltage recovery time penalty, the voltage oscillation penalty, the voltage lower bound, and the voltage recovery time upper bound. Since a company usually purchases devices via deferred payment and the devices will be replaced after their economic life time, the acquisition cost and the installation cost are represented by the annualized equivalent cost. The operating cost measures the cost of losses which is computed via TOPF. The voltage deviation penalty, the voltage recovery time penalty, and the voltage oscillation penalty are the components of the performance penalties. They are converted to corresponding

soft constraints that have a monetary value. In addition, some severe phenomena are prohibited, such as voltage recovery time exceeding upper limit and voltage below lower limit. For example, the voltage recovery times in this case cannot be more than two seconds and voltage magnitudes cannot be lower than 90% of their nominal values. The optimization problem avoids these phenomena by introducing hard constraints, which should be necessarily satisfied. Hard constraints are defined as follows:

Definition 7.1: Hard constraints represent absolute limitations imposed on the problem. In this problem, these constraints are the upper bound of voltage recovery time and are low bounds of bus voltages in the power system.

Dynamic programming is a very efficient method to solve decision problems. For a decision with n stages (exclude Stage 0) and m states in each stage, there are m^n routes from Stage 0 to Stage n . For example, the problem in Figure 7.2 has 20 stages and 16 states in each stage, so there are $16^{20} = 1.2089 \times 10^{24}$ routes. That means 1.2089×10^{24} trials are needed to find the minimum cost if an algorithm traverses all these routes. However, dynamic programming needs only m computations for each state at one stage. Therefore, the number of the computations is reduced to nm^2 . For the problem in this section, only $20 \times 16^2 = 5120$ computations are needed. The computational burden of dynamic programming is reduced to $1/(2.3612 \times 10^{20})$ of the trivial method.

A decision problem that can be solved by dynamic programming must satisfy two properties: optimal substructure and overlapping subproblem. Optimal substructure means that the optimal solution to the problem contains the optimal solutions to its subproblems. If a problem has the property of overlapping subproblem, a recursive algorithm should revisit the same problem repeatedly [129]. Therefore, the problem can be broken down into several reusable subproblems.

$C^*(X_{i,k})$ denotes the optimal trajectory cost from Stage 0 to State i at Stage k represented in Formula (7-12). Under this condition, $C^*(X_{j,k-1})$ is also optimal, otherwise we can replace it by the optimal one and obtain a smaller $C^*(X_{i,k})$. Therefore, the optimal

trajectory cost from Stage 0 to $X_{i,k}$ is the minimum of the optimal trajectory cost from Stage 0 to each state at Stage $k-1$ plus the cost of $X_{i,k}$ if there is a feasible transition.

According to the definition in Formula (7-12), the problem of computing $C^*(X_{i,k})$ is broken down into computing $C^*(X_{j,k-1})$ and $C(X_{i,k})$, both of which can be pre-computed and stored in a table for reuse. Thus, reactive planning problem exhibits overlapping subproblems. In Lemma 7-1, we prove that the reactive source planning problem is a dynamic programming problem.

Lemma 7-1: the reactive source planning problem is a dynamic programming problem.

Proof:

This proof consists of three parts:

1. The condition of dynamic programming is the principle of optimality.
2. Define the reactive source planning problem.
3. The reactive source planning problem satisfies the principle of optimality.

Part 1: The condition of dynamic programming is the principle of optimality.

Dynamic programming requires a decision problem satisfying the principle of optimality proposed by Bellman in 1953: an optimal policy has the property that whatever the initial state and the initial decision are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision [130]. This definition can be interpreted by Bellman's recursive equation [131] as follows:

For State i at Stage k , the optimal policy is given by

$$f^*(X_{i,k}) = \min_{all\ X_{i,k} \rightarrow X_{j,k+1}} \{T(X_{i,k} \rightarrow X_{j,k+1})C(X_{i,k}) + f^*(X_{j,k+1})\}, \quad (7-11)$$

where

$X_{i,k}$ = State i at Stage k ,

$X_{j,k+1}$ = State j at Stage $k+1$,

$T(X_{i,k} \rightarrow X_{j,k+1})$ = the cost of the decision $X_{i,k} \rightarrow X_{j,k+1}$ given State i at Stage k ($= 1$ when transition is feasible, $= +\infty$ when transition is infeasible) ,

$$C(X_{i,k}) = \text{the cost of State } i \text{ at Stage } k = J_{\text{AnnualizedEquivalentCost}}(X_{i,k}) + J_{\text{OperatingCost}}(X_{i,k}) \\ + J_{\text{VoltageDeviationPenalty}}(X_{i,k}) + J_{\text{VoltageRecoveryTimePenalty}}(X_{i,k}) + J_{\text{VoltageOscillationPenalty}}(X_{i,k}) \\ + J_{\text{VoltageLowerBound}}(X_{i,k}) + J_{\text{VoltageRecoveryTimeUpperBound}}(X_{i,k}),$$

$J_{\text{AnnualizedEquivalentCost}}(X_{i,k})$ = the annualized equivalent cost depending on the installed reactive sources and their economic life time,

$J_{\text{OperatingCost}}(X_{i,k})$ = the operating cost computed using TOPF,

$J_{\text{VoltageDeviationPenalty}}(X_{i,k})$ = the penalty associated with the voltage deviation of load buses from their nominal values, at steady state,

$J_{\text{VoltageRecoveryTimePenalty}}(X_{i,k})$ = the penalty associated with the time lengths after a fault clearance at which load buses reach 90% of their nominal values,

$J_{\text{VoltageOscillationPenalty}}(X_{i,k})$ = the penalty associated with the magnitudes of the voltage oscillation after a fault clearance,

$J_{\text{VoltageLowerBound}}(X_{i,k})$ = the hard constraint that the voltage deviation of load buses larger than 90% of their nominal values, at steady state,

$J_{\text{VoltageRecoveryTimeUpperBound}}(X_{i,k})$ = the hard constraint that the recovery time lengths are smaller than 2 seconds, and

$f^*(X_{i,k})$ = the optimal trajectory cost from $X_{i,k}$ to Stage m , where m = the total number of the stages.

According to Bellman's optimal policy and recursive Equation (7-11), a multi-stage decision problem can be solved by dynamic programming if the accumulated optimal trajectory cost from Stage k to Stage m equals the minimum or the maximum of the accumulated optimal trajectory cost from $X_{j,k+1}$ to Stage m plus the decision cost $X_{i,k} \rightarrow X_{j,k+1}$ among all candidate decisions of initial state $X_{i,k}$.

Part 2: Definition of reactive source planning problem.

The reactive source planning problem is defined as follows: the optimal trajectory cost from Stage 0 to Stage k equals the minimum of the optimal trajectory cost from

Stage 0 to Stage $k-1$ plus the cost of the decision $X_{j,k-1} \rightarrow X_{i,k}$ among all candidate decisions of $X_{j,k-1}$. The mathematical definition is as follows:

$$C^*(X_{i,k}) = \min_{\text{all state } X_{j,k-1} \text{ in stage } k-1} [C^*(X_{j,k-1}) + T(X_{j,k-1} \rightarrow X_{i,k})C(X_{i,k})], \quad (7-12)$$

where

$C^*(X_{i,k})$ = the optimal trajectory cost from Stage 0 to State i at Stage k ,

$C(X_{i,k})$ = the cost of State i at Stage k , and

$T(X_{j,k-1} \rightarrow X_{i,k})$ = the transition cost from $X_{j,k-1}$ to $X_{i,k}$.

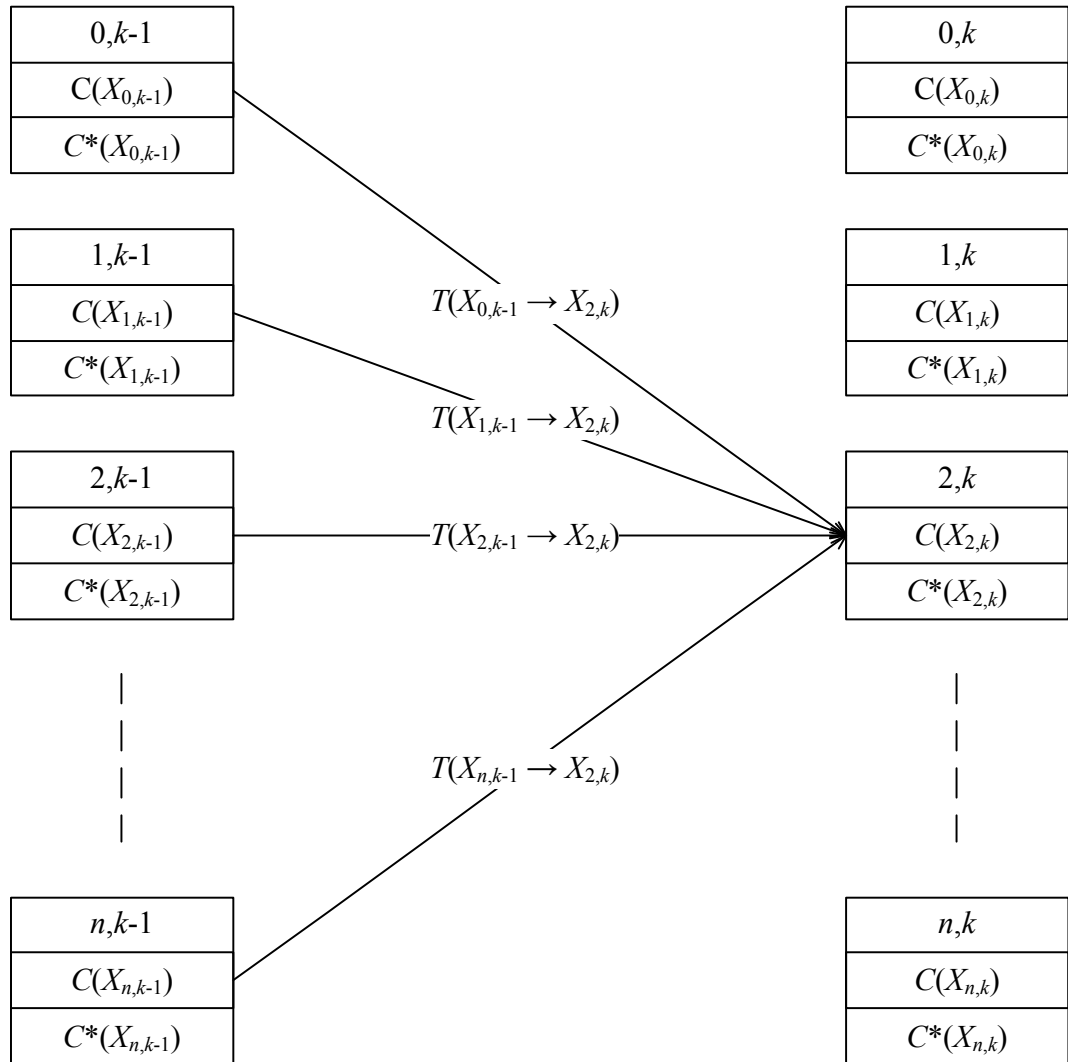


Figure 7.3 Transitions from Stage $k-1$ to Stage k

Figure 7.3 illustrates the transition procedure of Formula (7-12).

Part 3: Reactive source planning problem satisfies the principle of optimality.

We prove Part 3 by contradiction. Without loss of generality, the total number of stages is m and the number of states in each stage is n . An additional dummy node with the zero state cost after the final stage has been added to the decision tree to simplify the proof in Figure 7.4. The transition from any state in Stage m to the dummy node is feasible, that is, $T(X_{i,m} \rightarrow X_{dummy,m+}) = 1$ ($i = 0, \dots, n-1$).

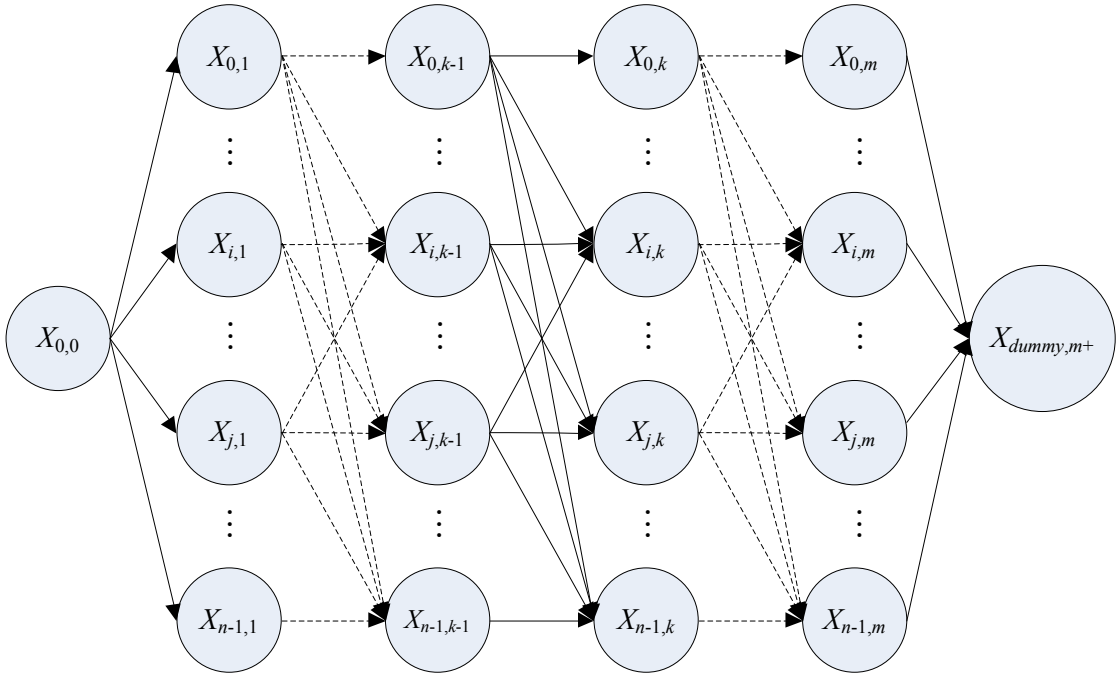


Figure 7.4 Decision tree: at each stage, there are n states and m stages

For the problem in Figure 7.4, $C^*(X_{dummy,m+})$ obviously equals the optimal trajectory cost of the original reactive planning problem in Formula (7-12) at the final stage. For an arbitrary state $X_{j,k-1}$, assume that the path $P(X_{j,k-1}, X_{dummy,m+})$ is the optimal path from $X_{j,k-1}$ to $X_{dummy,m+}$ and $X_{i,k} \in P(X_{j,k-1}, X_{dummy,m+})$. The decision process for the evaluation of the path $P(X_{j,k-1}, X_{dummy,m+})$ starts at $X_{j,k-1}$. This means that the nodes of

Stage 0 to $(k-1)$ are irrelevant to this decision process. Therefore, we assume optimal trajectory costs of all nodes before Stage k are zero. That is,

$$C^*(X_{p,q}) = 0, \text{ where } 0 \leq p \leq n-1 \text{ and } q < k. \quad (7-13)$$

If the reactive planning problem is a dynamic programming problem, $P(X_{i,k}, X_{dummy,m+}) \subset P(X_{j,k-1}, X_{dummy,m+})$ is the optimal path from $X_{i,k}$ to $X_{dummy,m+}$; otherwise, we assume $P'(X_{i,k}, X_{dummy,m+})$ is the optimal path from $X_{i,k}$ to $X_{dummy,m+}$.

Since $P(X_{j,k-1}, X_{dummy,m+})$ and $P'(X_{i,k}, X_{dummy,m+})$ are both optimal, we have

the cost of $P(X_{j,k-1}, X_{dummy,m+})$: $f^*(X_{j,k-1})$,

the cost of $P'(X_{i,k}, X_{dummy,m+})$: $f^{*'}(X_{i,k})$, and

the cost of $P(X_{i,k}, X_{dummy,m+})$:

$$\begin{aligned} f^*(X_{i,k}) &= f^*(X_{j,k-1}) - T(X_{j,k-1} \rightarrow X_{i,k})C(X_{i,k}) \\ &= C^*(X_{dummy,m+}) - C^*(X_{j,k-1}) - T(X_{j,k-1} \rightarrow X_{i,k})C(X_{i,k}). \\ &= C^*(X_{dummy,m+}) - T(X_{j,k-1} \rightarrow X_{i,k})C(X_{i,k}) \end{aligned} \quad (7-14)$$

Since $P'(X_{i,k}, X_{dummy,m+})$ is optimal, $f^{*'}(X_{i,k}) < f^*(X_{i,k})$. That is

$$f^{*'}(X_{i,k}) < C^*(X_{dummy,m+}) - T(X_{j,k-1} \rightarrow X_{i,k})C(X_{i,k}). \quad (7-15)$$

Then, we can select path $P'(X_{j,k-1}, X_{dummy,m+}) = X_{j,k-1} \rightarrow X_{i,k} \rightarrow P'(X_{i,k}, X_{dummy,m+})$ with cost equal to $f^{*'}(X_{i,k}) + T(X_{j,k-1} \rightarrow X_{i,k}) \times C(X_{i,k}) < C^*(X_{dummy,m+})$ according to (7-15). This contradicts with the assumption that $P(X_{j,k-1}, X_{dummy,m+})$ is optimal. Therefore, the reactive source planning problem satisfies the principle of optimality. ■

The state with the lowest cost at the final stage is the optimal final state, and the lowest cost route to that state is the optimal planning schedule. That is,

$$\min_{\text{all state } X_{i,m} \text{ in stage } m} C^*(X_{i,m}), \quad (7-16)$$

where

m = the number of the stages to be planned (= 20 in the problem in Figure 7.2).

7.6 The Algorithm Structure

The dynamic programming algorithm traverses all stages and finally obtains the optimal trajectory cost from Stage 0 to Stage m . Figure 7.5 shows the architecture of the algorithm.

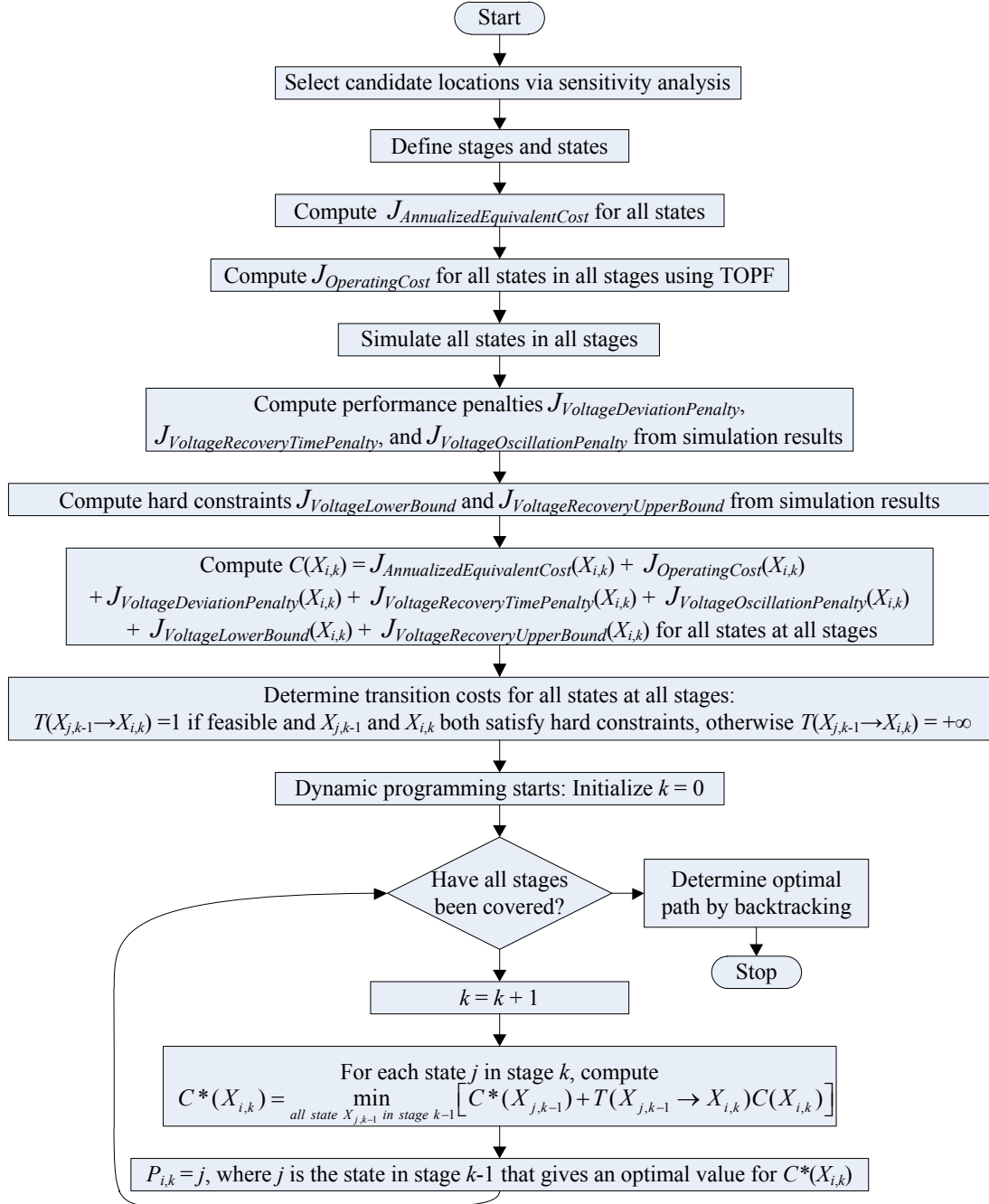


Figure 7.5 The reactive planning algorithm

7.7 Summary

This chapter introduced the optimal VAR allocation problem and its solution method via dynamic programming. Several costs used in the problem are defined at the beginning: the annualized equivalent cost, the operating cost, the voltage deviation penalty, the voltage recovery penalty, the voltage oscillation penalty, the hard constraint of voltage magnitudes, and the hard constraint of voltage recovery time. The annualized equivalent cost is computed from the investment cost and the installation cost of the VAR devices. The operating cost is computed by TOPF proposed in Chapter 6. Other costs are obtained from simulation results. The planning algorithm starts from selecting candidate locations via sensitivity analysis on voltage deviation indices. The locations with larger negative index slopes are selected. Then, the optimization problem was defined and the proof is given that the problem can be solved using dynamic programming. Finally, the algorithm structure was presented.

CHAPTER 8

DEMONSTRATION AND EVALUATION OF PROPOSED OPF METHOD WITH SEVERAL TEST SYSTEMS

8.1 Introduction

This chapter demonstrates the OPF algorithm by several cases: a three-bus system, the RTS-79 system, the RTS-96 system, and several other test systems of size up to 300 buses. We implemented the OPF algorithm using Visual C++ and ran it on an Intel Core2 Duo CPU SP9400 (6M Cache, 2.40 GHz, 1066 MHz FSB) with 8GB memory.

8.2 A Three-Bus System Example

The three-bus system is illustrated in Figure 8.1 [73]. The unit data for this system are given in Table 8-1.

Table 8-1: Unit parameters in actual units in the three-bus system

	a (\$/h)	b (\$/MW·h)	c (\$/MW ² ·h)	P_{\max} (MW)	P_{\min} (MW)	Q_{\max} (MVar)	Q_{\min} (MVar)
Unit #1	102	12	0.01	100.0	11.0	50.0	-20.0
Unit #2	180	10	0.02	150.0	15.0	40.0	-25.0
Unit #3	95	13	0.01	75.0	8.0	30.0	-20.0

The power base is assumed to be 100MW, and the unit data in the per-unit scale are shown in Table 8-2.

Table 8-2: Unit parameters in per-unit scale in the three-bus system

	a (\$/h)	b (\$/pu·h)	c (\$/pu ² ·h)	P_{\max} (pu)	P_{\min} (pu)	Q_{\max} (pu)	Q_{\min} (pu)
Unit #1	102	1,200	100	1.0	0.11	0.5	-0.2
Unit #2	180	1,000	200	1.5	0.15	0.4	-0.25
Unit #3	95	1,300	100	0.75	0.08	0.3	-0.2

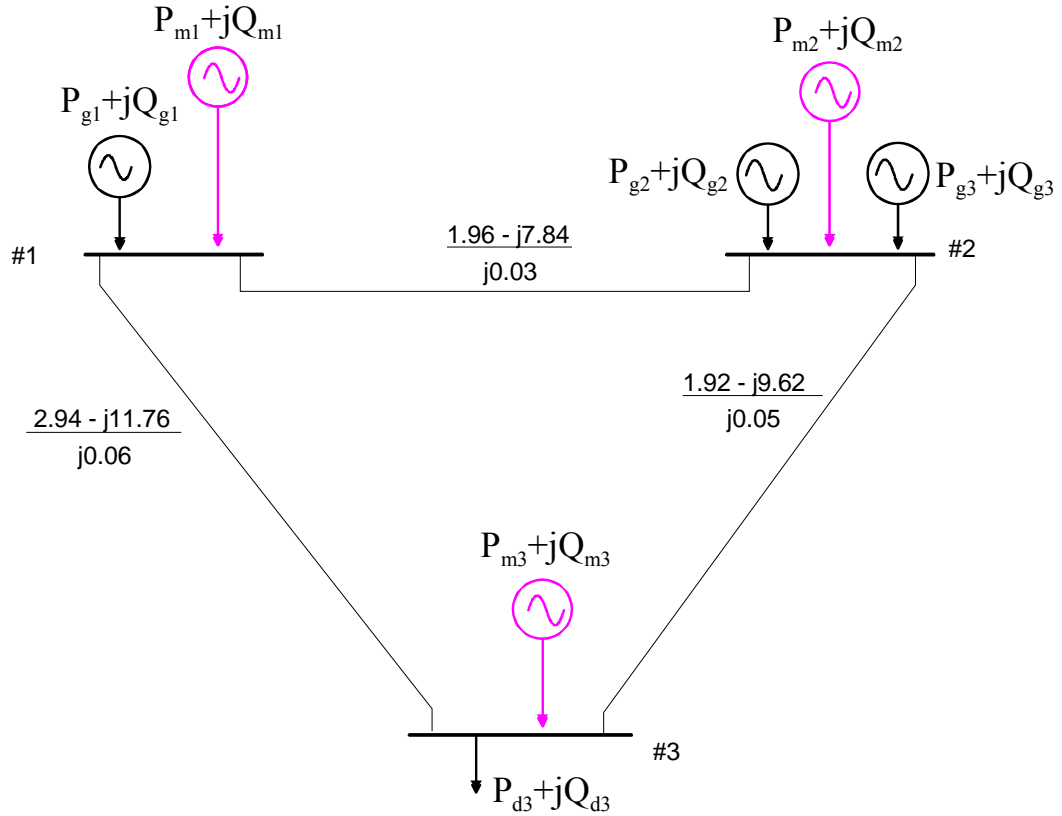


Figure 8.1 A three-bus power system

The electric load at Bus 3 is $2.0 + j0.58 \text{ pu}$. The initial values of the control variables are set to: $P_{g1} = 0.8 \text{ pu}$, $P_{g2} = 0.9 \text{ pu}$, and $P_{g3} = 0.3 \text{ pu}$. The initial operating state is assumed to be: $V_1 = V_2 = V_3 = 1.0 \text{ pu}$ and $\delta_1 = \delta_2 = \delta_3 = 0.0$. The voltage constraints are $0.99 \leq V_1 \leq 1.01$, $0.97 \leq V_2 \leq 1.03$, and $0.97 \leq V_3 \leq 1.03$. The penalty factor μ is selected to be 10,000 \$/(pu·hour). No linearization limit is added.

Chapter 3 shows the quadratic power system model. In addition, a power system can be modeled via the polar model, which uses voltage magnitudes and phases instead of real and imaginary voltages. The polar model forms the power flow equations using real and reactive power balance equations, while the quadratic model forms the power flow equations using the real and reactive power balance equations or the real and imaginary current conservation equations. The quadratic model usually uses the current conservation equations since their order are equal to or less than the power balance

equations. This OPF algorithm works for both models with different mismatch sources, where the polar model uses the real and reactive power sources and the quadratic model uses the real and imaginary current sources. This three-bus example shows the algorithm procedure via both the polar and quadratic models for demonstration. This section gives only the nonlinear optimization problem in the first iteration and the final result for concision.

8.2.1 The Solution with Polar Power Flow

8.2.1.1 The Nonlinear Optimization Problem at the First Iteration

The objective function:

$$\text{Minimize } \mu(1-\nu) \sum_{k=1}^3 \left(|P_{mk}^o| + |Q_{mk}^o| \right) + \sum_{k=1}^3 \sum_{j=1}^{M(k)} \left(a_{k,j} + b_{k,j} P_{gk,j} + c_{k,j} P_{gk,j}^2 \right). \quad (8-1)$$

Subject to the real power balance equation:

$$\begin{aligned} P(\mathbf{x}, \mathbf{u}) &= \sum P - P_L - q \\ &= P_{g1,1} + P_{g2,1} + P_{g2,2} + (1-\nu)(P_{m1}^o + P_{m2}^o + P_{m3}^o) - P_{d3} \\ &\quad - \left[4.9V_1^2 + 3.88V_2^2 + 4.86V_3^2 \right. \\ &\quad \left. - 3.92V_1V_2 \cos \delta_2 - 5.88V_1V_3 \cos \delta_3 - 3.84V_2V_3 \cos(\delta_3 - \delta_2) \right] \end{aligned} \quad (8-2)$$

The reactive power balance equation:

$$\begin{aligned} Q(\mathbf{x}, \mathbf{u}) &= \sum Q - Q_L - q \\ &= Q_{g1,1} + Q_{g2,1} + Q_{g2,2} + (1-\nu)(Q_{m1}^o + Q_{m2}^o + Q_{m3}^o) - Q_{d3} \\ &\quad - \left[-19.51V_1^2 - 17.38V_2^2 - 21.27V_3^2 + 15.68V_1V_2 \cos \delta_2 \right. \\ &\quad \left. + 23.52V_1V_3 \cos \delta_3 + 19.24V_2V_3 \cos(\delta_3 - \delta_2) \right] \end{aligned} \quad (8-3)$$

The operating constraints:

Empty,

The control variable constraints:

$$0 \leq \nu \leq 1, \quad 0.99 \leq V_1 \leq 1.01, \quad 0.15 \leq P_{g2} \leq 1.5, \quad 0.08 \leq P_{g3} \leq 0.75,$$

$$0.0 \leq \delta_1 \leq 0.0, \quad -0.25 \leq Q_{g2} \leq 0.4, \quad -0.2 \leq Q_{g3} \leq 0.3,$$

The control variables:

$$v \text{ and } \mathbf{u} = [V_1 \quad \delta_1 \quad P_{g2,1} \quad Q_{g2,1} \quad P_{g2,2} \quad Q_{g2,2}]^T,$$

The state variables:

$$[P_{g1,1} \quad Q_{g1,1} \quad V_2 \quad \delta_2 \quad V_3 \quad \delta_3]^T.$$

8.2.1.2 Solution Report and Analysis

The potential operating constraints to be added to the model are $0.1 \leq P_{g1,1} \leq 1.0$, - $0.2 \leq Q_{g1,1} \leq 0.5$, $0.97 \leq V_2 \leq 1.03$, and $0.97 \leq V_3 \leq 1.03$. In this example, the operating constraints $V_2 \leq 1.03$ and $P_{g1,1} \leq 1.0$ are added at the second iteration and $0.97 \leq V_3$ is add at the third iteration. $Step_{mismatch}$ is set to 3, while the actual steps used are 5 due to the linearization error. The final solution is

$$[P_{g1,1}^o \quad Q_{g1,1}^o]^T = [0.464 \quad 0.21]^T,$$

$$[P_{g2,1}^o \quad Q_{g2,1}^o]^T = [1.5 \quad 0.1738]^T,$$

$$[P_{g2,2}^o \quad Q_{g2,2}^o]^T = [0.08 \quad 0.1213]^T,$$

$$[V_1^o \quad \delta_1^o]^T = [1.01 \quad 0.0]^T,$$

$$[V_2^o \quad \delta_2^o]^T = [1.0221 \quad 0.0468]^T, \text{ and}$$

$$[V_3^o \quad \delta_3^o]^T = [0.97 \quad -0.0646]^T.$$

The optimal cost is \$3018.4628/hour. Figure 8.2 shows that the cost with mismatch decreases before the mismatch variables reach zero, while Figure 8.3 shows that the cost without mismatch increases. The reason of that is loss and power transmission through the transmission lines is small when the mismatch values are large. Figure 8.4 presents the iteration steps of the real powers, the reactive powers, the voltage magnitudes, the voltage angles, the real power mismatches, and the reactive power mismatches.

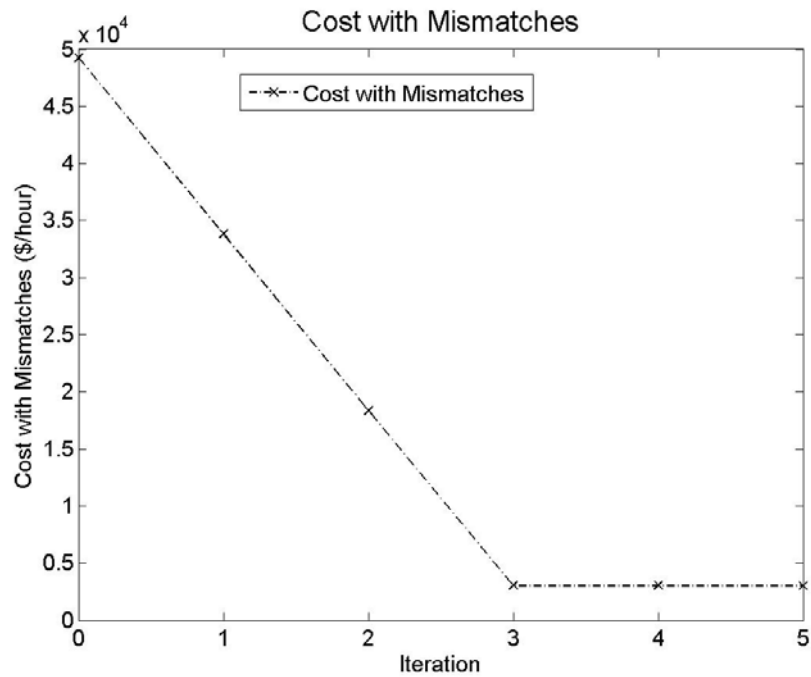


Figure8.2 The cost with mismatches for the three-bus system using the polar power flow

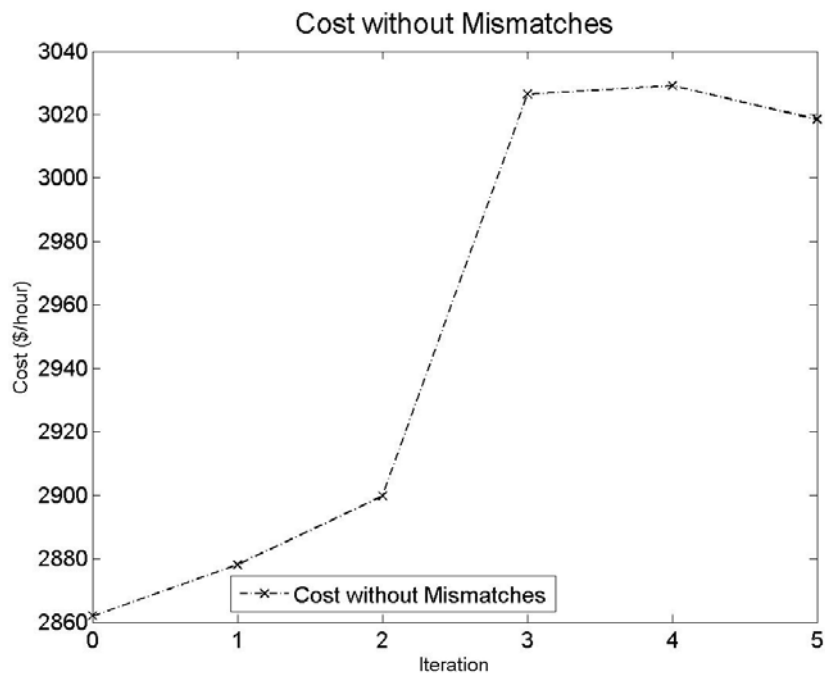


Figure 8.3 The cost without mismatches for the three-bus system using the polar power flow

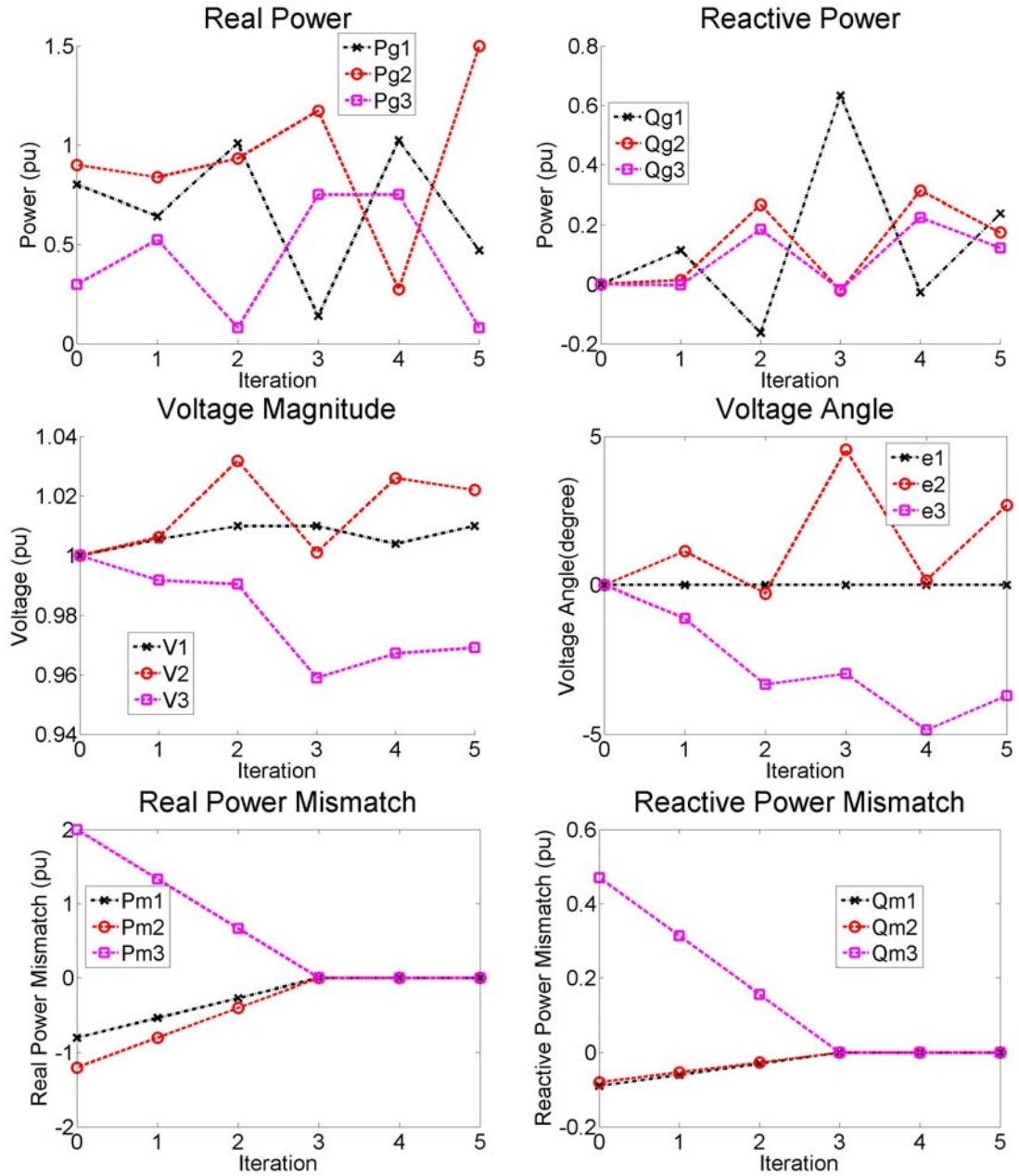


Figure 8.4 The variables of the three-bus system using the polar power flow

8.2.2 The Solution with Quadratized Power Flow

This example uses the power balance equations in the optimization problem for demonstration since they are quadratic when the system does not consist of any

transformers. Otherwise, the quadratic model may need more state variables to ensure that the power balance equations are quadratized.

8.2.2.1 Nonlinear Optimization Problem at the First Iteration

The objective function:

$$\text{Minimize } \mu(1-\nu) \sum_{k=1}^3 \left(\left| I_{mkr}^o \right| + \left| I_{mki}^o \right| \right) + \sum_{k=1}^3 \sum_{j=1}^{M(k)} \left(a_{k,j} + b_{k,j} P_{gk,j} + c_{k,j} P_{gk,j}^2 \right). \quad (8-4)$$

Subject to the real power balance equation:

$$\begin{aligned} P(\mathbf{x}, \mathbf{u}) &= \sum P - P_L - q \\ &= P_{g1,1} + P_{g2,1} + P_{g2,2} - 4.9V_{1r}^2 - 3.88(V_{2r}^2 + V_{2i}^2) - 3.92(V_{3r}^2 + V_{3i}^2) \\ &\quad + 3.92V_{1r}V_{2r} + 5.88V_{1r}V_{3r} + 3.84(V_{2r}V_{3r} + V_{2i}V_{3i}) - P_{d3} \\ &\quad + (1-\nu)(V_{1r}I_{m1r}^o + V_{1i}I_{m1i}^o + V_{2r}I_{m2r}^o + V_{2i}I_{m2i}^o + V_{3r}I_{m3r}^o + V_{3i}I_{m3i}^o) \end{aligned} \quad (8-5)$$

The reactive power balance equation:

$$\begin{aligned} Q(\mathbf{x}, \mathbf{u}) &= \sum Q - Q_L - q \\ &= Q_{g1,1} + Q_{g2,1} + Q_{g2,2} + 19.51V_{1r}^2 + 17.38(V_{2r}^2 + V_{2i}^2) + 21.27(V_{3r}^2 + V_{3i}^2) \\ &\quad - 15.68V_{1r}V_{2r} - 23.52V_{1r}V_{3r} - 19.24(V_{2r}V_{3r} + V_{2i}V_{3i}) - Q_{d3} \\ &\quad + (1-\nu)(V_{1r}I_{m1i}^o + V_{1i}I_{m1r}^o + V_{2r}I_{m2i}^o + V_{2i}I_{m2r}^o + V_{3r}I_{m3i}^o + V_{3i}I_{m3r}^o) \end{aligned} \quad (8-6)$$

The operating constraints:

Empty,

The control variable constraints:

$$0 \leq \nu \leq 1, \quad 0.99 \leq V_1 \leq 1.01, \quad 0.15 \leq P_{g2} \leq 1.5, \quad 0.08 \leq P_{g3} \leq 0.75,$$

$$0.0 \leq \delta_1 \leq 0.0, \quad -0.25 \leq Q_{g2} \leq 0.4, \quad -0.2 \leq Q_{g3} \leq 0.3,$$

The control variables:

$$\nu \text{ and } \mathbf{u} = \begin{bmatrix} V_{1r} & V_{1i} & P_{g2,1} & Q_{g2,1} & P_{g2,2} & Q_{g2,2} \end{bmatrix}^T,$$

The state variables:

$$\begin{bmatrix} P_{g1,1} & Q_{g1,1} & V_{1mag} & E_{g1,1r} & E_{g1,1i} \end{bmatrix}^T,$$

$$\begin{bmatrix} V_{2r} & V_{2i} & V_{2mag} & E_{g2,1r} & E_{g2,1i} & E_{g2,2r} & E_{g2,2i} \end{bmatrix}^T,$$

$$\begin{bmatrix} V_{3r} & V_{3i} & V_{3mag} & u_{d31} & u_{d32} \end{bmatrix}^T.$$

8.2.2.2 Solution Report and Analysis

The potential operating constraints to be added to the model are $0.1 \leq P_{g1,1} \leq 1.0$, - $0.2 \leq Q_{g1,1} \leq 0.5$, $0.97 \leq V_{2mag} \leq 1.03$, and $0.97 \leq V_{3mag} \leq 1.03$. In this example, the operating constraint $V_{2mag} \leq 1.03$ is added at the second iteration and $0.97 \leq V_{3mag}$ is add at the third iteration. $Step_{Mismatch}$ is set to 3, while the actual steps used are 4 due to the linearization error. The final solution is:

$$\begin{bmatrix} P_{g1,1}^o & Q_{g1,1}^o \end{bmatrix}^T = [0.471 \quad 0.0641]^T,$$

$$\begin{bmatrix} P_{g2,1}^o & Q_{g2,1}^o \end{bmatrix}^T = [1.5 \quad 0.276]^T,$$

$$\begin{bmatrix} P_{g2,2}^o & Q_{g2,2}^o \end{bmatrix}^T = [0.08 \quad 0.1951]^T,$$

$$\begin{bmatrix} V_{1r}^o & V_{1i}^o & V_{1mag}^o \end{bmatrix}^T = [1.0042 \quad 0.0 \quad 1.01]^T,$$

$$\begin{bmatrix} V_{2r}^o & V_{2i}^o & V_{2mag}^o \end{bmatrix}^T = [1.0277 \quad 0.0446 \quad 1.0287]^T, \text{ and}$$

$$\begin{bmatrix} V_{3r}^o & V_{3i}^o & V_{3mag}^o \end{bmatrix}^T = [0.967 \quad -0.0642 \quad 0.969]^T.$$

The binding constraints in the solution using the quadratized power flow are fewer than the solution using the polar power flow and the variable values in these two solutions are slightly different because of linearization differences. In these two solutions, the real powers are closer than the reactive powers since they are more related to the cost. The final cost using the quadratized power flow is \$3018.9908/hour, which is very close to the final cost using the polar power flow. Figure 8.5 and Figure 8.6 show the operating costs with and without the mismatch penalties respectively.

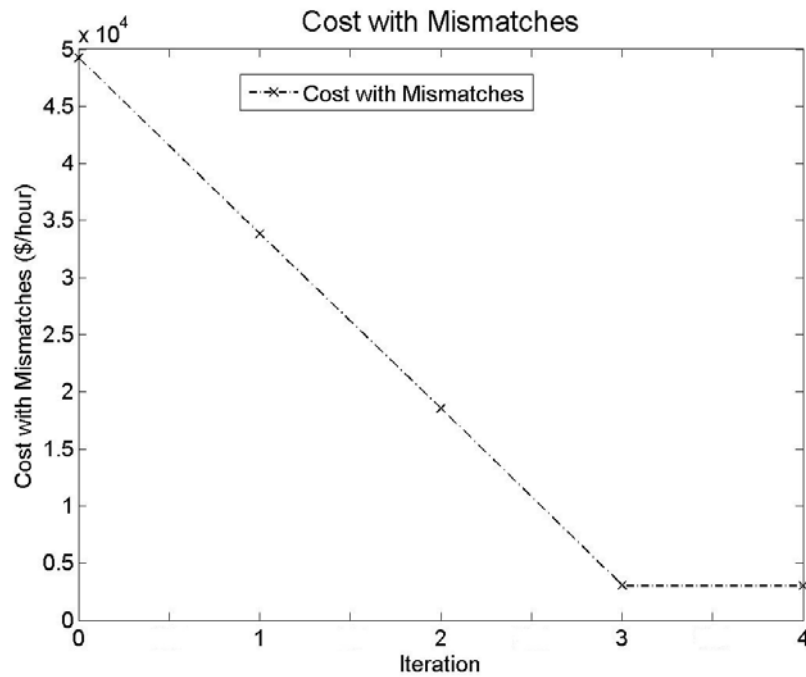


Figure 8.5 The cost with mismatches for the three-bus system using the quadratized power flow

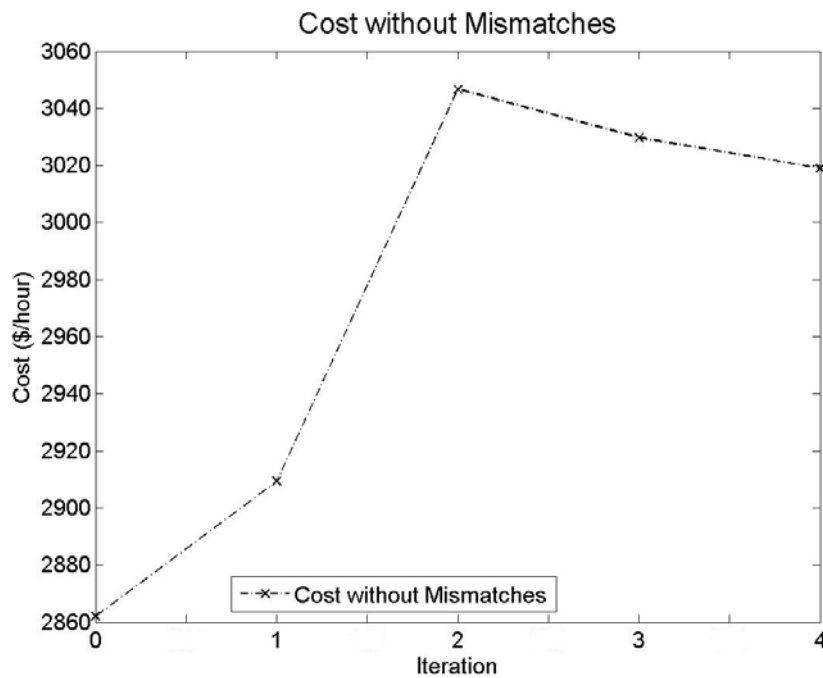


Figure 8.6 The cost without mismatches for the three-bus system using the quadratized power flow

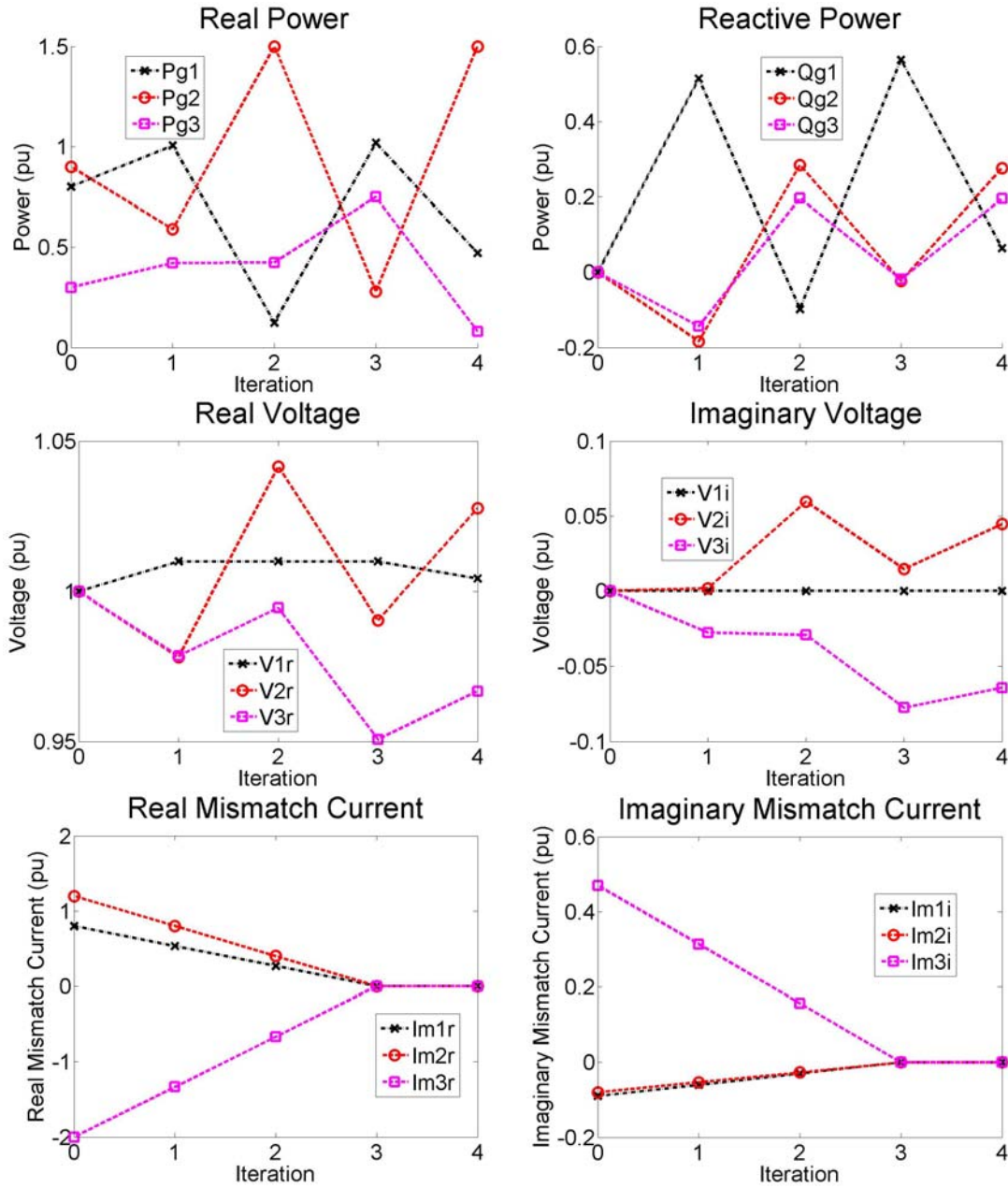


Figure 8.7 Variables for the three-bus system using the quadratized power flow

Figure 8.7 presents the iteration steps of the real powers, the reactive powers, the real voltages, the imaginary voltages, the real power mismatches, and the reactive power mismatches. Since the derivatives of the objective function with respect to the reactive powers are very small, the iteration scenarios of the reactive powers are much different

from the case using the polar power flow. In addition, the solution with different reactive powers can have the same cost.

8.3 The RTS-79 System Example

The RTS-79 system is from [76] and also Zone 1 in [77] shown in Figure 8.8. This system has 24 buses and 2 voltage levels $138kV$ and $230kV$ connected through tap-control transformers.

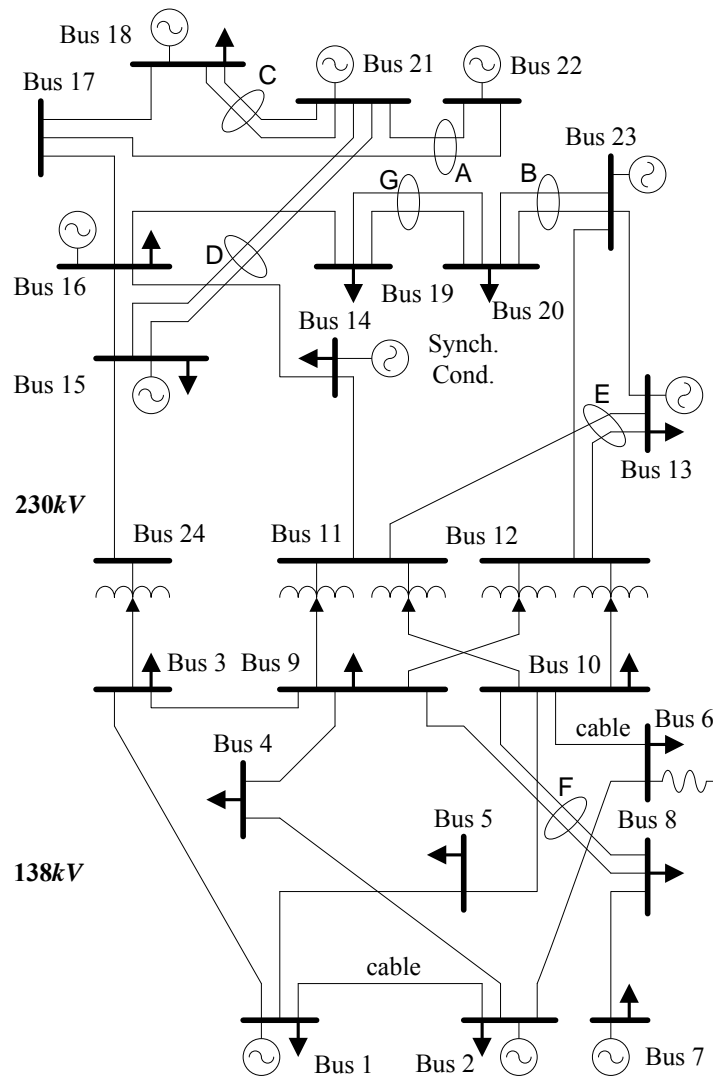


Figure 8.8 The IEEE RTS-79 power system

The system parameters are obtained from Table 1, 7, and 12 in [77]. The cost function of the j th unit at Bus k is $c_{k,j}(\mathbf{x}, \mathbf{u}) = a_{k,j} + b_{k,j}P_{k,j} + c_{k,j}P_{k,j}^2$, where the coefficients are computed via linear regression using the data in Table 9 in [77]. The coefficients in the cost function are also converted to the per-unit scale from the actual unit MW since the algorithm runs at the per-unit scale. Hydro units are irrelevant to the optimization process and usually run at maximum possible capacities in a heavy loaded power system since their $b_{k,j}$ and $c_{k,j}$ are zero. The formula to compute the cost coefficients for other types of units is

$$\begin{bmatrix} a_{k,j} \\ b_{k,j} \\ c_{k,j} \end{bmatrix} = (H^T H)^{-1} H^T b + \begin{bmatrix} a_{k,j,OM} \\ b_{k,j,OM} \\ 0 \end{bmatrix}, \quad (8-7)$$

where H is the matrix of heat rate. The number of the columns in H and b equals the number of the real power levels in Table 9 in [77]. For a real power level P , the column in H is $[1 \ P \ P^2]^T$ and the column in b is its corresponding cost. $a_{k,j,OM}$ and $b_{k,j,OM}$ are the coefficients of the operation and maintenance costs respectively [90].

The OPF algorithm plans to reduce the mismatch variables to zero in 5 iterations and uses the current conservation equations instead of the power balance equations. The penalty factor μ is set to 10^7 . The bus voltages are limited between $0.95pu$ and $1.05pu$ and the transformer taps are between $0.9pu$ and $1.1pu$. Table 7 in [77] gives the constraints of the real and reactive powers. Table 12 in [77] gives the constraints of the transmission lines and the transformers.

Figure 8.9 and Figure 8.10 show the costs in each iteration. The cost with mismatch reduces linearly as the setup, and the one more iteration is used due to nonlinearity. Figure 8.11 shows the real power loss in each iteration. Figure 8.11 looks similar to Figure 8.10 since the operating cost is positive correlated to the loss. The more real power generated, the more loss occurs.

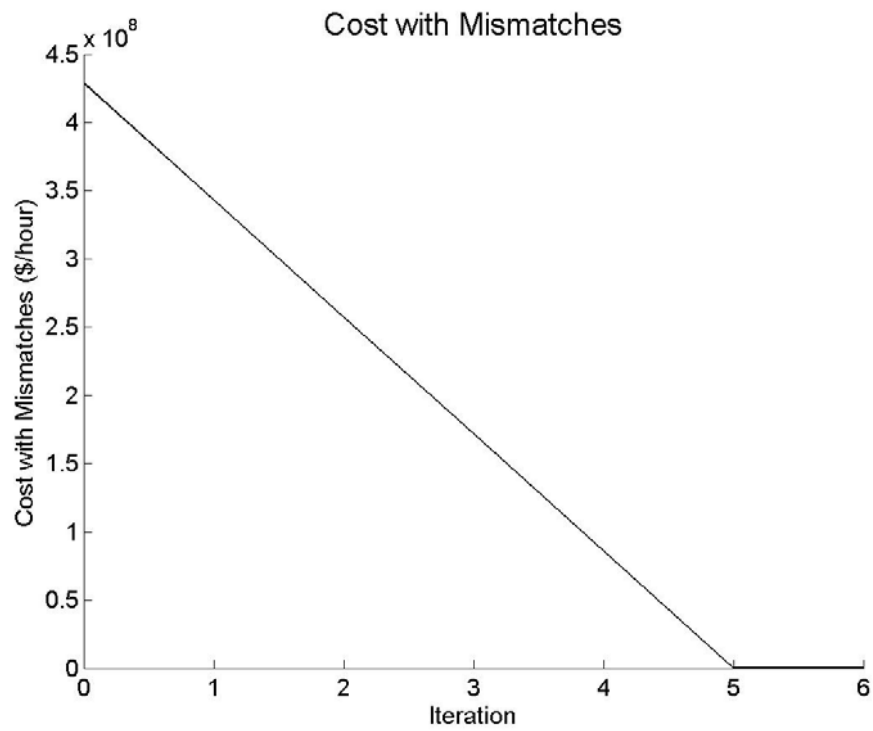


Figure 8.9 The cost with mismatches for the RTS-79 system

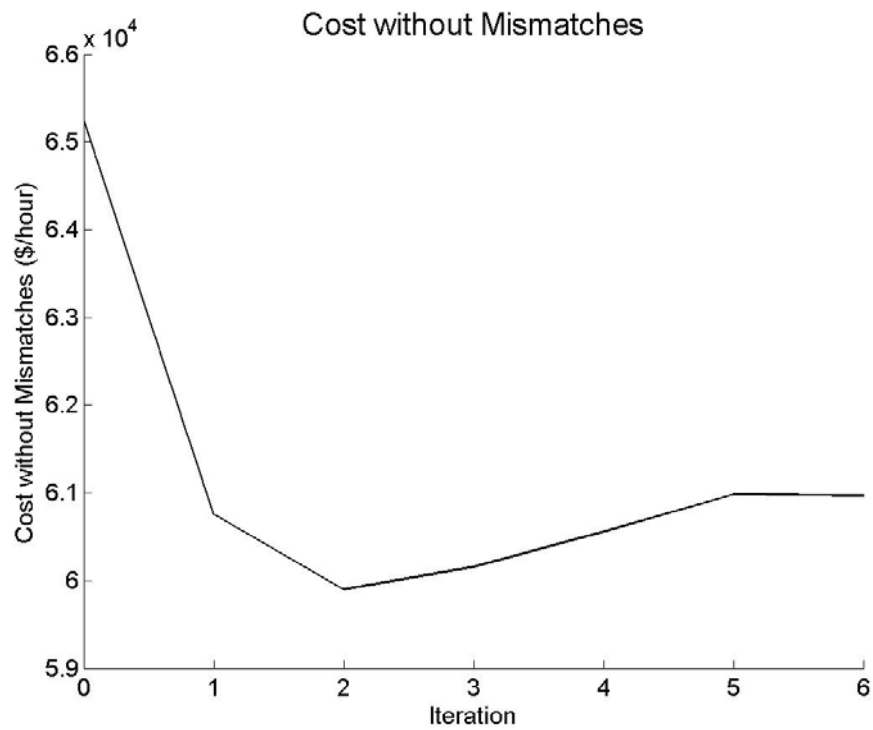


Figure 8.10 The cost without mismatches for the RTS-79 system

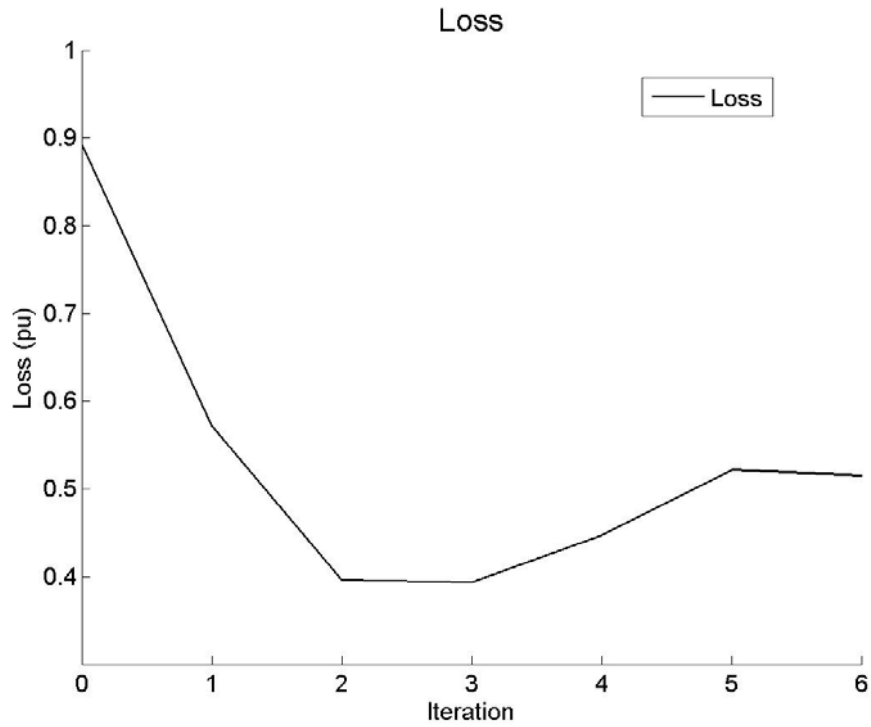


Figure 8.11 The real power loss of the RTS-79 system

According to the algorithm setup, the model includes six operating constraints initially. Table 8-3 shows the operating constraints added from Iteration 1. The number of the potential operating constraints is 90, which = $2 \times \# \text{ the buses} + \# \text{ the transmission lines} + \# \text{ the transformers} + 2 \times \# \text{ the slack mode generators}$. The actual number of the operating constraints added is 33, around 1/3 of the total number.

Table 8-4 presents the runtime information without parallelism. The time unit is second. PF stands for the number of the power flow iterations. The first column lists the loop indices. Three numbers are assigned to three loop layers shown in Figure 5.1. For example, 4-2-1 represents the 4th Layer-1 loop, the 2nd Layer-2 loop, and the 1st Layer-3 loop. Each Layer-3 loop includes three steps: linearization, linear programming, and solving the power flow. Their runtime information is shown from the third to the fifth columns. The last column shows the iteration numbers used to solve the power flow. The

last row of each Layer-1 loop gives its total runtime. The total runtime of the whole algorithm is 0.09124 seconds. The following inequities hold:

the linearization time in Loop 2-1-1 > the linearization time in Loop 2-2-1 > the linearization time in Loop 2-2-2.

The reason is that Loop 2-1-1 linearizes all the operating constraints (# = 31), Loop 2-2-1 linearizes the additional operating constraints (# = 2), and Loop 2-2-2 updates only violated the modeled linearized operating constraints. Loop 1-1-1 has the longest power flow runtime and the maximum iteration number since the move lengths of the control variables are longer at the beginning of the algorithm.

Table 8-3: The operating constraints added at each iteration for the RTS-79 system

#	Operating constraints
1	$0.95^2 \leq V_{101r}^2 + V_{101i}^2, 0.95^2 \leq V_{102r}^2 + V_{102i}^2, 0.95^2 \leq V_{103r}^2 + V_{103i}^2,$ $0.95^2 \leq V_{104r}^2 + V_{104i}^2, 0.95^2 \leq V_{105r}^2 + V_{105i}^2, 0.95^2 \leq V_{106r}^2 + V_{106i}^2,$ $0.95^2 \leq V_{107r}^2 + V_{107i}^2, 0.95^2 \leq V_{108r}^2 + V_{108i}^2, 0.95^2 \leq V_{109r}^2 + V_{109i}^2,$ $0.95^2 \leq V_{110r}^2 + V_{110i}^2, 0.95^2 \leq V_{111r}^2 + V_{111i}^2, 0.95^2 \leq V_{112r}^2 + V_{112i}^2,$ $0.95^2 \leq V_{114r}^2 + V_{114i}^2, 0.95^2 \leq V_{115r}^2 + V_{115i}^2, 0.95^2 \leq V_{116r}^2 + V_{116i}^2,$ $0.95^2 \leq V_{117r}^2 + V_{117i}^2, 0.95^2 \leq V_{118r}^2 + V_{118i}^2, 0.95^2 \leq V_{119r}^2 + V_{119i}^2,$ $0.95^2 \leq V_{120r}^2 + V_{120i}^2, 0.95^2 \leq V_{121r}^2 + V_{121i}^2, 0.95^2 \leq V_{122r}^2 + V_{122i}^2,$ $0.95^2 \leq V_{123r}^2 + V_{123i}^2, 0.95^2 \leq V_{124r}^2 + V_{124i}^2$
2	$V_{101r}^2 + V_{101i}^2 \leq 1.05^2, V_{102r}^2 + V_{102i}^2 \leq 1.05^2$
3	None
4	$V_{107r}^2 + V_{107i}^2 \leq 1.05^2$
5	None
6	$V_{122r}^2 + V_{122i}^2 \leq 1.05^2$

The contingencies include PV/PQ generator outages, transmission line outages, and transformer outages. The number of the contingencies in the RTS-79 system is 71. This OPF software is run on each post-contingency case. OPF results show that five of those do not have any feasible solution. That means these contingencies are the most severe in the system. They are Generator 23 at Bus 18, Generator 24 at Bus 21, Generator

33 at Bus 23, Transmission Line 10 between Bus 6 and Bus 10, and Transmission Line 11 between Bus 7 and Bus 8.

Table 8-4: Runtime information of the RTS-79 system without parallelism

#	Layer-3 Runtime	Linearization Runtime	LP Runtime	Power Flow Runtime	PF
1-1-1	0.00649	0.00147	0.00062	0.00330	6
1-2-1	0.00766	0.00411	0.00137	0.00215	5
1	0.01418				
2-1-1	0.00902	0.00475	0.00116	0.00217	5
2-2-1	0.00390	0.00042	0.00126	0.00219	5
2-2-2	0.00352	0.00005	0.00127	0.00218	5
2	0.01651				
3-1-1	0.00935	0.00536	0.00095	0.00216	5
3-1-2	0.00318	0.00005	0.00097	0.00214	5
3-1-3	0.00319	0.00005	0.00097	0.00215	5
3	0.01584				
4-1-1	0.00903	0.00508	0.00091	0.00215	5
4-1-2	0.00307	0.00005	0.00086	0.00213	5
4-2-1	0.00346	0.00024	0.00105	0.00214	5
4-2-2	0.00316	0.00005	0.00094	0.00214	5
4	0.01877				
5-1-1	0.00908	0.00513	0.00094	0.00216	5
5-1-2	0.00324	0.00005	0.00102	0.00214	5
5	0.01234				
6-1-1	0.00871	0.00536	0.00076	0.00173	4
6-2-1	0.00289	0.00025	0.00088	0.00174	4
6	0.01166				

Table 8-5 presents the runtime information with parallelism on a dual-core CPU. Loop 2-1-1 linearizes all the operating constraints in 0.00277s compared with 0.00477s without parallelism. The runtime is reduced to around one half. Parallelism affects runtime mainly on the loops with more operating constraints. The total runtime of the whole algorithm is 0.07764s. Parallelism reduces 14.91% of the total runtime.

Table 8-5: Runtime information of the RTS-79 system with parallelism

#	Layer-3 Runtime	Linearization Runtime	LP Runtime	Power Flow Runtime	PF
1-1-1	0.00587	0.00138	0.00063	0.00275	6
1-2-1	0.00778	0.00385	0.00135	0.00256	5
1	0.01369				
2-1-1	0.00704	0.00277	0.00118	0.00219	5
2-2-1	0.00386	0.00041	0.00125	0.00217	5
2-2-2	0.00391	0.00005	0.00122	0.00262	5
2	0.01488				
3-1-1	0.00679	0.00269	0.00099	0.00219	5
3-1-2	0.00324	0.00005	0.00097	0.00219	5
3-1-3	0.00323	0.00005	0.00099	0.00217	5
3	0.01328				
4-1-1	0.00602	0.00161	0.00132	0.00217	5
4-1-2	0.00307	0.00005	0.00084	0.00216	5
4-2-1	0.00348	0.00027	0.00100	0.00217	5
4-2-2	0.00322	0.00005	0.00096	0.00218	5
4	0.01585				
5-1-1	0.00641	0.00225	0.00095	0.00233	5
5-1-2	0.00327	0.00005	0.00100	0.00219	5
5	0.00970				
6-1-1	0.00555	0.00217	0.00080	0.00173	4
6-2-1	0.00289	0.00025	0.00090	0.00173	4
6	0.00850				

8.4 The RTS-96 System Example

The RTS-96 system is three RTS-79 systems with several long-distance connections [77]. The problem setup is the same as in the RTS-79 system. The costs in each iteration are shown in Figure 8.12 and Figure 8.13. Figure 8.14 presents the real power loss in each iteration. An observation is that the actual cost decreases at the fifth and the sixth iteration, while the real power loss increases at these two iterations since less cost does not mean less loss. Table 8-6 shows the operating constraints added in each iteration. The total number of the operating constraints is 272, while the actual number of the operating constraints added is 45 in total. The number of the operating constraints

added at the first iteration is much larger since the changes of the variable values are larger.

The ideal phenomenon of the algorithm is that the cost without mismatch reduces tremendously at the first iteration and increases a little bit due to more constraints included. This phenomenon is the same as those in the three-bus system and in the RTS-79 system. However, the cost without mismatch in the RTS-96 system reduces monotonously, because the linearization bounds limit the step lengths of the control variables. Therefore, the convergence speed of larger systems will be limited by the linearization error. On the other hand, large linearization bounds may result in no power flow solution.

Table 8-7 presents the runtime information without parallelism and Table 8-8 presents the runtime information with parallelism. The time unit is second. The total runtime without parallelism is 0.860637 seconds and the total runtime with parallelism is 0.735704 seconds.

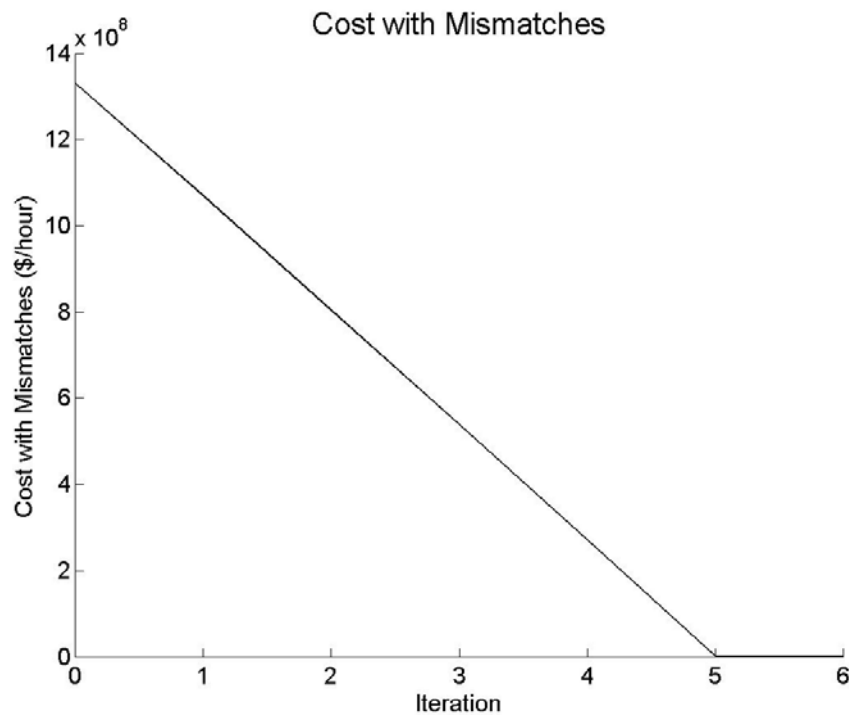


Figure 8.12 The cost with mismatches for the RTS-96 system

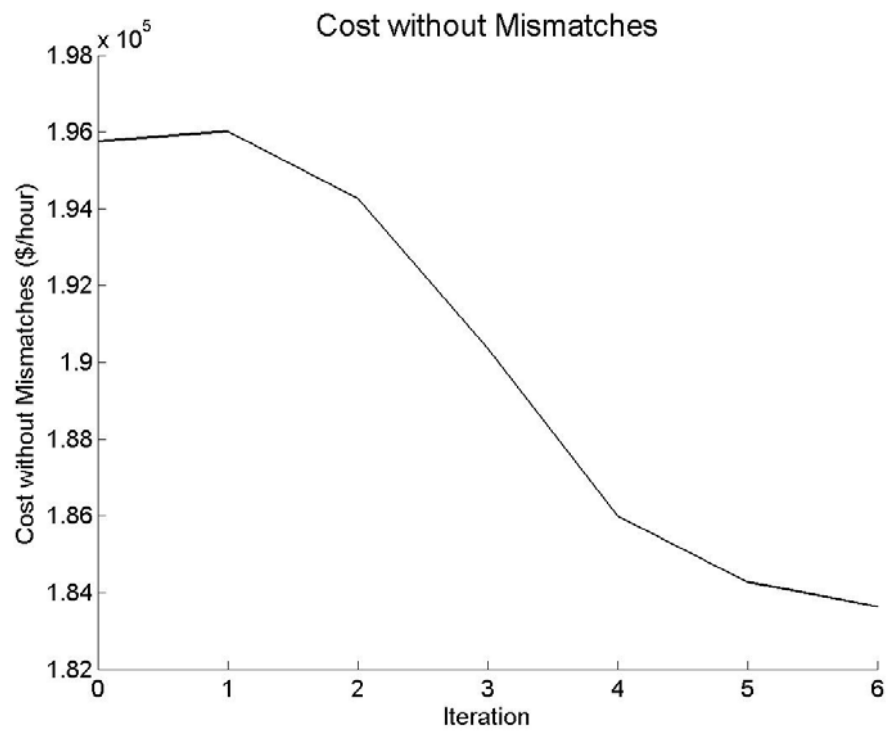


Figure 8.13 The cost without mismatches for the RTS-96 system

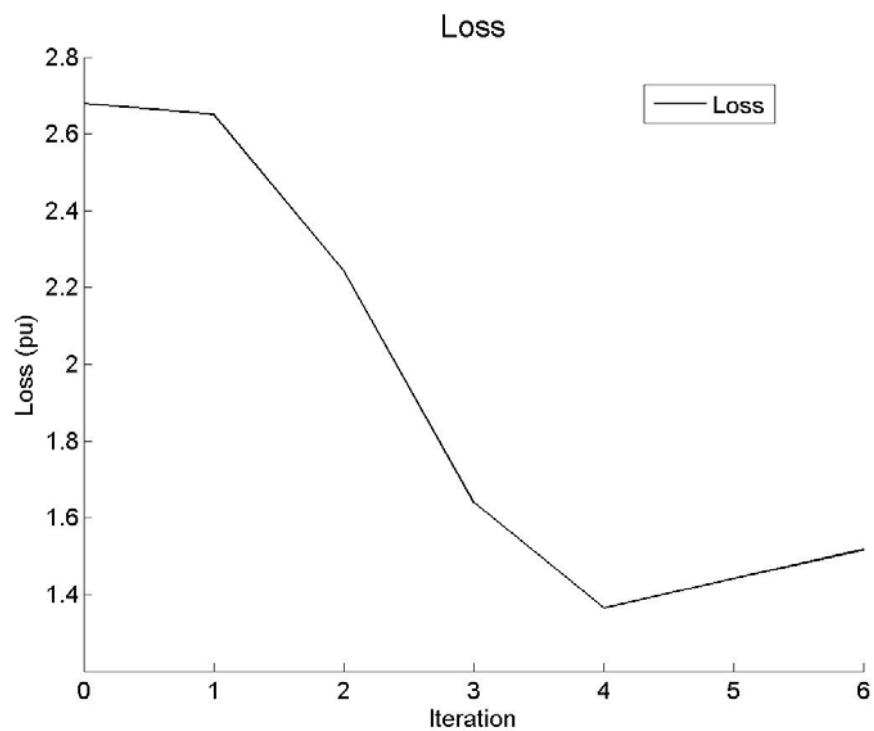


Figure 8.14 The real power loss of the RTS-96 system

Table 8-6: The operating constraints added at each iteration for the RTS-96 system

#	Operating constraints
1	$ \begin{aligned} &V_{101r}^2 + V_{101i}^2 \leq 1.05^2, V_{102r}^2 + V_{102i}^2 \leq 1.05^2, V_{117r}^2 + V_{117i}^2 \leq 1.05^2, \\ &V_{118r}^2 + V_{118i}^2 \leq 1.05^2, V_{121r}^2 + V_{121i}^2 \leq 1.05^2, V_{122r}^2 + V_{122i}^2 \leq 1.05^2, \\ &0.95^2 \leq V_{208r}^2 + V_{208i}^2, V_{301r}^2 + V_{301i}^2 \leq 1.05^2, V_{302r}^2 + V_{302i}^2 \leq 1.05^2, \\ &V_{303r}^2 + V_{303i}^2 \leq 1.05^2, V_{304r}^2 + V_{304i}^2 \leq 1.05^2, V_{305r}^2 + V_{305i}^2 \leq 1.05^2, \\ &V_{306r}^2 + V_{306i}^2 \leq 1.05^2, V_{307r}^2 + V_{307i}^2 \leq 1.05^2, V_{308r}^2 + V_{308i}^2 \leq 1.05^2, \\ &V_{309r}^2 + V_{309i}^2 \leq 1.05^2, V_{310r}^2 + V_{310i}^2 \leq 1.05^2, V_{311r}^2 + V_{311i}^2 \leq 1.05^2, \\ &V_{312r}^2 + V_{312i}^2 \leq 1.05^2, V_{313r}^2 + V_{313i}^2 \leq 1.05^2, V_{314r}^2 + V_{314i}^2 \leq 1.05^2, \\ &V_{315r}^2 + V_{315i}^2 \leq 1.05^2, V_{316r}^2 + V_{316i}^2 \leq 1.05^2, V_{317r}^2 + V_{317i}^2 \leq 1.05^2, \\ &V_{318r}^2 + V_{318i}^2 \leq 1.05^2, V_{319r}^2 + V_{319i}^2 \leq 1.05^2, V_{320r}^2 + V_{320i}^2 \leq 1.05^2, \\ &V_{321r}^2 + V_{321i}^2 \leq 1.05^2, V_{322r}^2 + V_{322i}^2 \leq 1.05^2, V_{323r}^2 + V_{323i}^2 \leq 1.05^2, \\ &V_{324r}^2 + V_{324i}^2 \leq 1.05^2, V_{325r}^2 + V_{325i}^2 \leq 1.05^2 \end{aligned} $
2	$V_{201r}^2 + V_{201i}^2 \leq 1.05^2, V_{202r}^2 + V_{202i}^2 \leq 1.05^2, V_{207r}^2 + V_{207i}^2 \leq 1.05^2$
3	None
4	None
5	$V_{218r}^2 + V_{218i}^2 \leq 1.05^2, V_{221r}^2 + V_{221i}^2 \leq 1.05^2, V_{222r}^2 + V_{222i}^2 \leq 1.05^2$
6	$V_{223r}^2 + V_{223i}^2 \leq 1.05^2$

The contingency number in the RTS-96 system is 219. OPF results show that four of those do not have any feasible solution. They are Transmission Line 10 between Bus 106 and Bus 110, Transmission Line 48 between Bus 206 and Bus 210, Transmission Line 86 between Bus 306 and Bus 310, and Transmission Line 120 between Bus 318 and Bus 223. The first three of those are at the same location in each zone, and the last one connects Zone 2 and Zone 3. The contingencies with no power flow solution do not include the generator outages since this system has more dispatch choices than the RTS-79 system does.

Table 8-7: Runtime information of the RTS-96 system without parallelism

#	Layer-3 Runtime	Linearization Runtime	LP Runtime	Power Flow Runtime	PF
1-1-1	0.028860	0.011207	0.001380	0.011779	7
1-2-1	0.059427	0.041676	0.006225	0.011443	7
1-2-2	0.018953	0.000183	0.006892	0.011810	7
1	0.107337				
2-1-1	0.066303	0.048126	0.005844	0.00876	5
2-1-2	0.014380	0.000183	0.005828	0.008304	5
2-1-3	0.014300	0.000182	0.005872	0.008184	5
2-2-1	0.018000	0.002967	0.006555	0.008427	5
2-2-2	0.015269	0.000193	0.006828	0.008185	5
2-2-3	0.014762	0.000193	0.006220	0.008288	5
2	0.143176				
3-1-1	0.070713	0.051147	0.008306	0.007651	5
3-1-2	0.015819	0.000184	0.007988	0.007584	5
3-1-3	0.016230	0.000184	0.007970	0.008017	5
3-1-4	0.016657	0.000192	0.008395	0.008009	5
3-2-1	0.023277	0.004317	0.010857	0.008049	5
3-2-2	0.018024	0.000201	0.010010	0.007750	5
3-2-3	0.017844	0.000199	0.009775	0.007809	5
3	0.178645				
4-1-1	0.075032	0.053234	0.010639	0.007792	5
4-1-2	0.017574	0.000200	0.009729	0.007579	5
4-1-3	0.018381	0.000200	0.010320	0.007801	5
4-1-4	0.019193	0.000209	0.010641	0.008282	5
4-2-1	0.020150	0.001776	0.010488	0.007833	5
4-2-2	0.018479	0.000205	0.010490	0.007720	5
4-2-3	0.018873	0.000219	0.010875	0.007714	5
4	0.187839				
5-1-1	0.075822	0.055812	0.008187	0.008406	5
5-1-2	0.017107	0.000216	0.008813	0.008001	5
5-1-3	0.016288	0.000216	0.007930	0.008079	5
5-1-4	0.016390	0.000216	0.008075	0.008037	5
5-1-5	0.017362	0.000215	0.008722	0.00836	5
5	0.143029				
6-1-1	0.078233	0.059373	0.006958	0.008371	5
6-1-2	0.017328	0.000217	0.008955	0.008086	5
6	0.095694				

Table 8-8: Runtime information of the RTS-96 system with parallelism

#	Layer-3 Runtime	Linearization Runtime	LP Runtime	Power Flow Runtime	PF
1-1-1	0.023727	0.006261	0.001325	0.011797	7
1-2-1	0.057903	0.040158	0.006432	0.011261	7
1-2-2	0.020041	0.000184	0.00699	0.012799	7
1	0.101762				
2-1-1	0.045657	0.027951	0.005799	0.008181	5
2-1-2	0.01577	0.000183	0.005973	0.009543	5
2-1-3	0.016242	0.000185	0.006299	0.009688	5
2-2-1	0.016794	0.001643	0.006685	0.008404	5
2-2-2	0.0153	0.000195	0.006392	0.008644	5
2-2-3	0.015258	0.000195	0.006293	0.008705	5
2	0.125194				
3-1-1	0.044509	0.023248	0.008951	0.008009	5
3-1-2	0.016987	0.000195	0.008723	0.008003	5
3-1-3	0.017275	0.000195	0.008877	0.008143	5
3-1-4	0.016913	0.000195	0.008416	0.008241	5
3-2-1	0.023599	0.004288	0.011273	0.007989	5
3-2-2	0.019581	0.000212	0.011293	0.008008	5
3-2-3	0.021117	0.000212	0.012674	0.008166	5
3	0.160062				
4-1-1	0.048882	0.025968	0.0109	0.008008	5
4-1-2	0.01874	0.000358	0.010103	0.008205	5
4-1-3	0.019318	0.000211	0.011035	0.008011	5
4-1-4	0.019403	0.000211	0.011174	0.007958	5
4-2-1	0.021087	0.001622	0.011074	0.008342	5
4-2-2	0.019467	0.00022	0.01107	0.008111	5
4-2-3	0.019445	0.000218	0.01121	0.007955	5
4	0.166514				
5-1-1	0.046422	0.026436	0.008216	0.008252	5
5-1-2	0.01719	0.000217	0.008543	0.008363	5
5-1-3	0.016315	0.000218	0.007744	0.008292	5
5-1-4	0.017697	0.000217	0.00797	0.009448	5
5-1-5	0.017228	0.000219	0.008823	0.008117	5
5	0.114911				
6-1-1	0.042454	0.024823	0.00766	0.006474	5
6-1-2	0.014517	0.000218	0.007798	0.006434	5
6	0.057103				

8.5 Test Systems with Different Sizes

This section presents the results of nine cases of different sizes: 6 buses [110], 9 buses [111], 14 buses [112], 24 buses [76], [77], 30 buses [113], 39 buses [114]-[117], 57 buses, 118 buses, and 300 buses [112]. Table 8-9 presents runtime information (in second) of these systems for several algorithms. Bold numbers mean that the algorithm does not converge at the specific case. MATLAB stands for MATLAB Optimization Toolbox. Successive LP stands for the sparse successive LP method. PDIPM stands for the prime/dual interior point method [118]. SC-PDIPM stands for the step-controlled variant of PDIPM. TRALM stands for the trust region based augmented Lagrangian method [117]. MIPS stands for MATLAB Interior Point Solver using the primal/dual point method. SC-MIPS stands for the step-controlled variant of MIPS [117], [120], [121]. SLP-sc stands for the proposed SLP algorithm on the solo-core platform. SLP-dc stands for the proposed SLP algorithm on the dual-core platform. SQP-sc stands for the proposed SQP algorithm on the solo-core platform. SQP-dc stands for the proposed SQP algorithm on the dual-core platform. Since MINOS [119] solver does not support 64-bit operating systems, this work uses PDIPM in the TSPOPF package instead of MINOS for comparison.

Table 8-9: Runtime information of nine cases of different sizes

	6	9	14	24	30	39	57	118	300
MATLAB	1.95	0.47	0.75	3.04	1.12	0.95	1.78	413.2	8329.55
Successive LP	1.04	0.73	1.06	4.95	5.4	13.81	4.73	8.89	31.67
PDIPM	0.04	0.03	0.04	0.06	0.07	0.07	0.09	0.19	0.53
SC-PDIPM	0.05	0.05	0.06	0.07	0.07	0.16	0.11	0.27	2.55
TRALM	0.06	0.05	0.14	0.24	0.19	0.97	0.23	4.27	8.91
MIPS	0.39	0.08	0.11	0.11	0.13	0.13	0.14	0.25	0.59
SC-MIPS	0.09	0.11	0.14	0.13	0.14	0.17	0.17	0.36	2.54
SLP-sc	0.0075	0.006	0.037	0.091	0.021	0.106	0.187	1.103	3.391
SLP-dc	0.0084	0.007	0.036	0.078	0.019	0.082	0.082	0.797	2.116
SQP-sc	0.039	0.020	0.083	0.364	0.065	0.328	0.386	2.719	4.755
SQP-dc	0.037	0.014	0.083	0.257	0.062	0.242	0.295	2.304	3.433

MATLAB runs very slow, especially for large-scale systems. For example, its runtime on the 300-bus system are more than two hours. Successive LP and TRALM do not converge in some cases since they need select appropriate parameters for each case and they may fall into local minimum when starting from an inappropriate working point. These phenomena usually occur on systems with large sizes and/or peak loads. The proposed algorithm is also a sequential algorithm, so it is very important to select appropriate values of the parameters where the linearization bound is the most important one. For example, a smaller bound may result in more iterations but a larger bound may result in an infeasible power flow solution. A future research orientation of this algorithm is a smarter value-selection method for the parameters. PDIPM and MIPS both use interior-point methods. Their runtimes are smaller, but they need feasible starting working points.

Figure 8.15 - Figure 8.23 present costs with/without mismatches in each iteration for all the cases in this section. Since the quadratized transformer model and the quadratized generator model are different from the corresponding polar models in the compared works, this work modifies these systems appropriately and adds several additional parameters, such as generator internal impedance and transformer magnetizing admittance. The curves in Figure 8.15 - Figure 8.23 are different, because these systems have different structures and parameters.

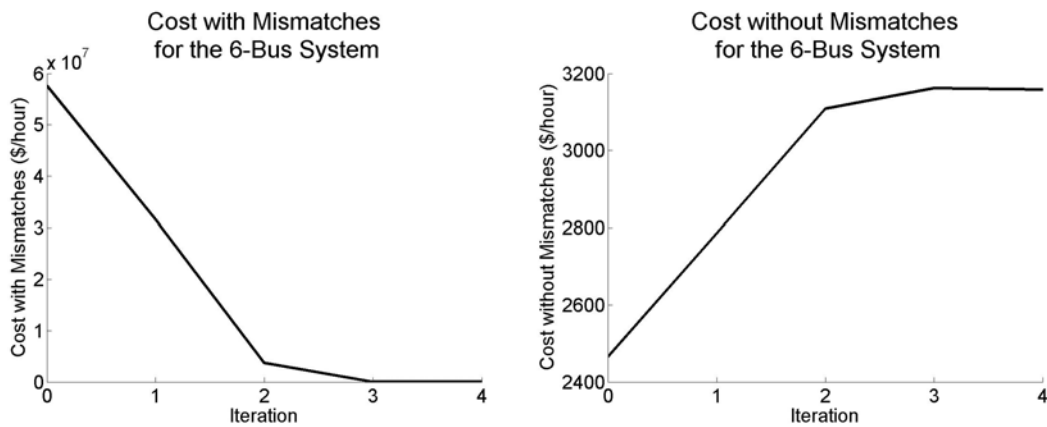


Figure 8.15 The costs with/without mismatches for the 6-bus system

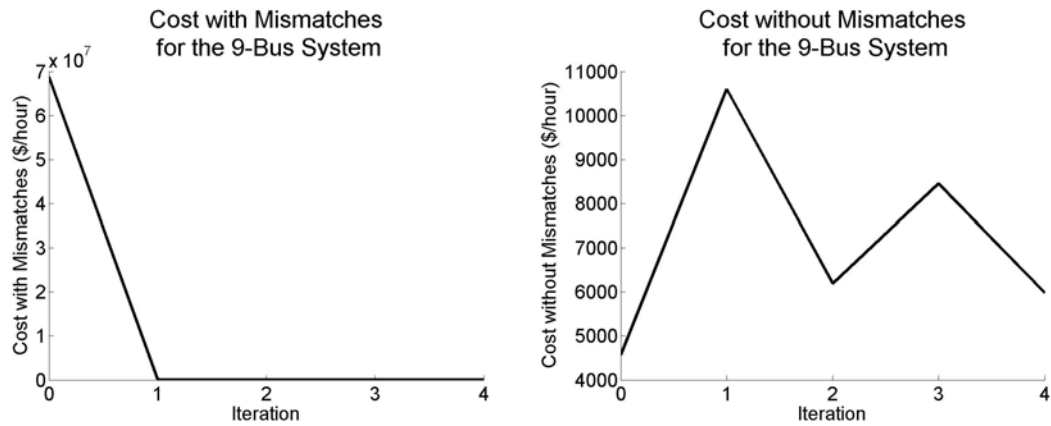


Figure 8.16 The costs with/without mismatches for the 9-bus system

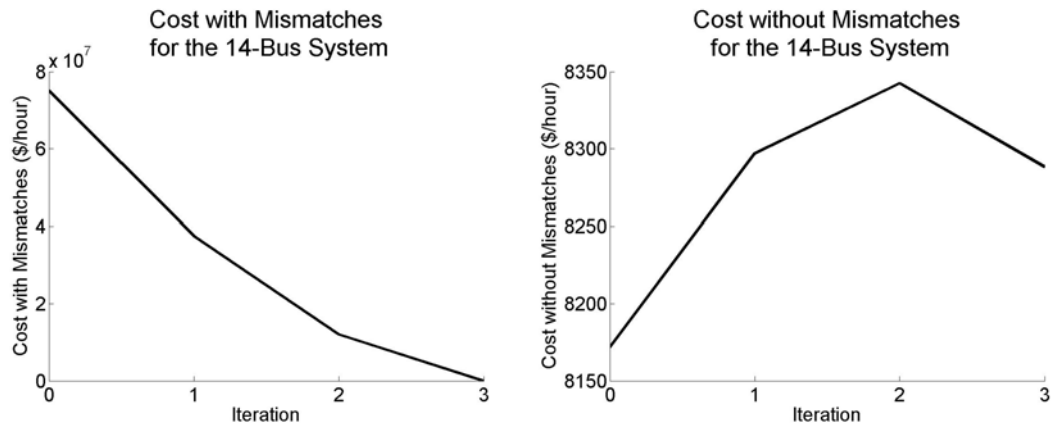


Figure 8.17 The costs with/without mismatches for the 14-bus system

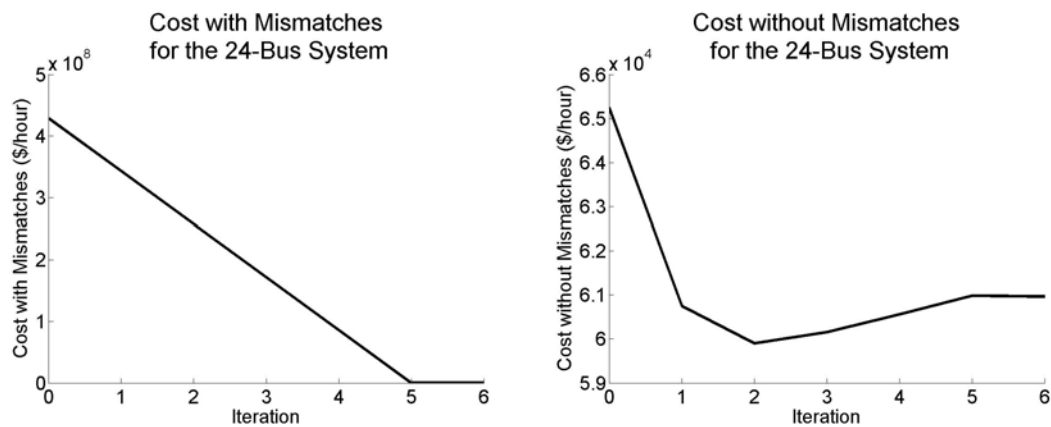


Figure 8.18 The costs with/without mismatches for the 24-bus system

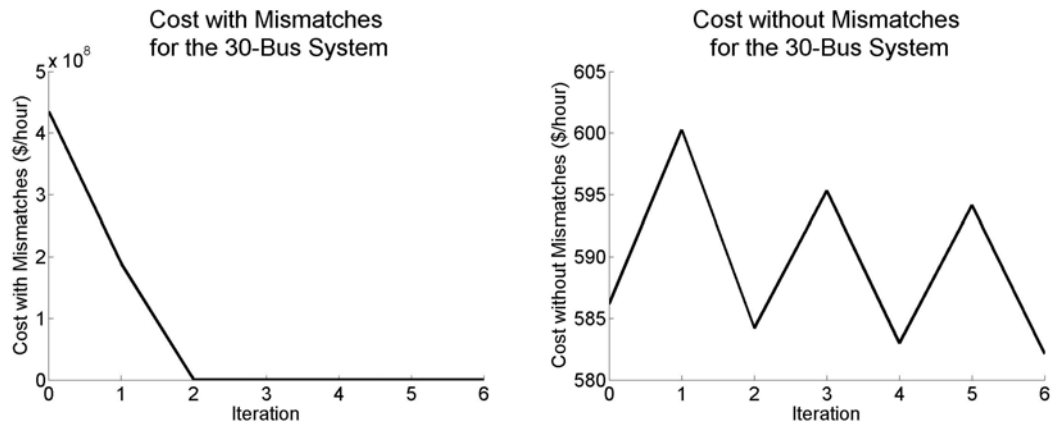


Figure 8.19 The costs with/without mismatches for the 30-bus system

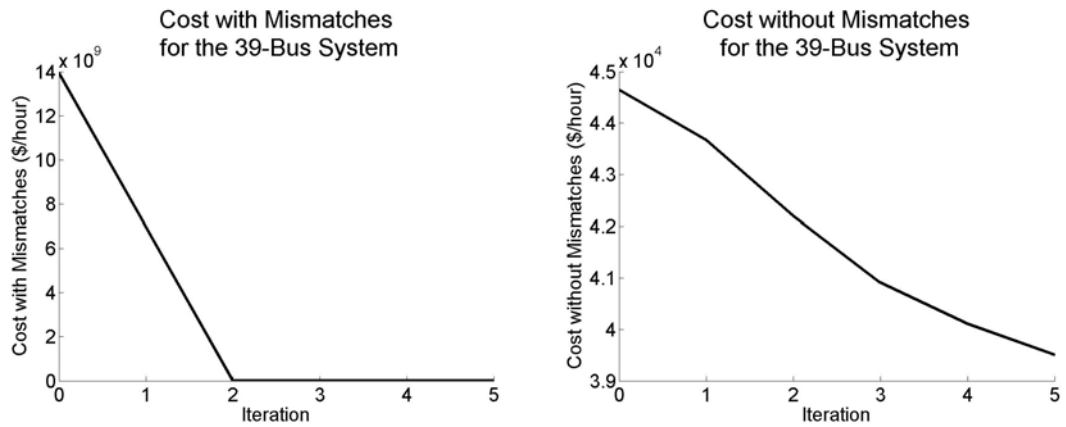


Figure 8.20 The costs with/without mismatches for the 39-bus system

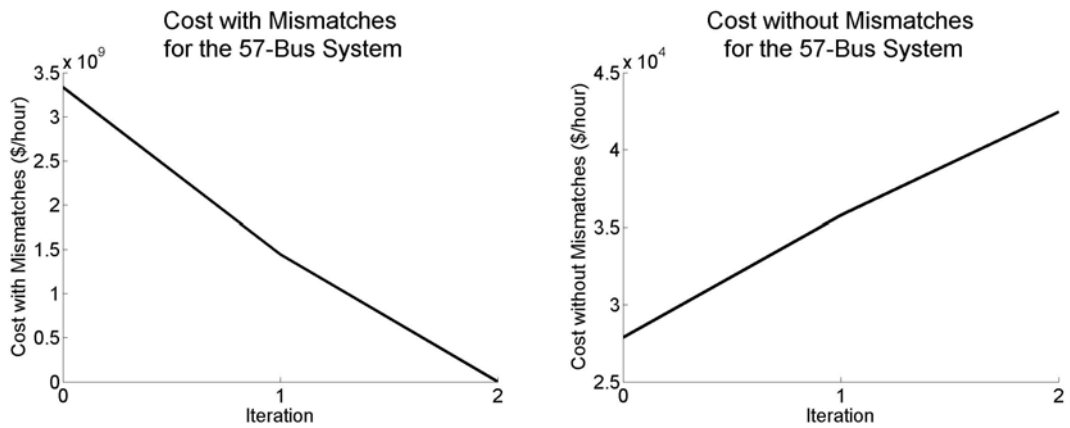


Figure 8.21 The costs with/without mismatches for the 57-bus system

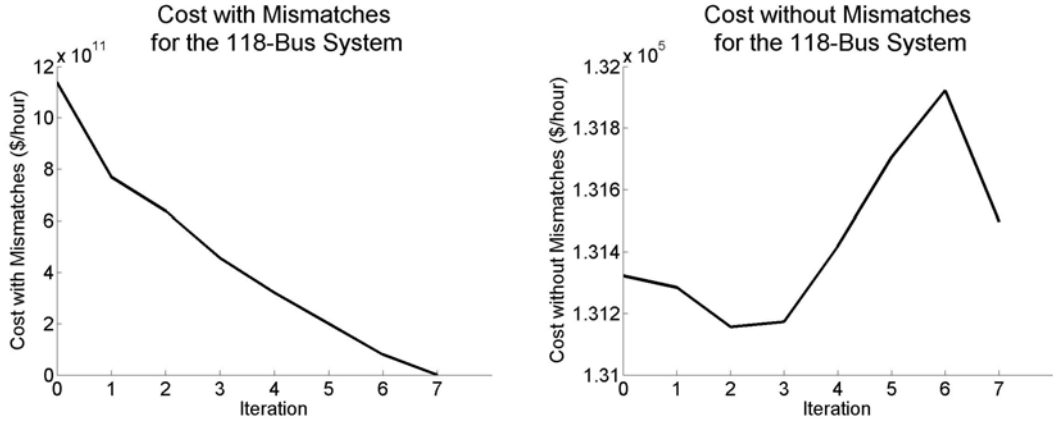


Figure 8.22 The costs with/without mismatches for the 118-bus system

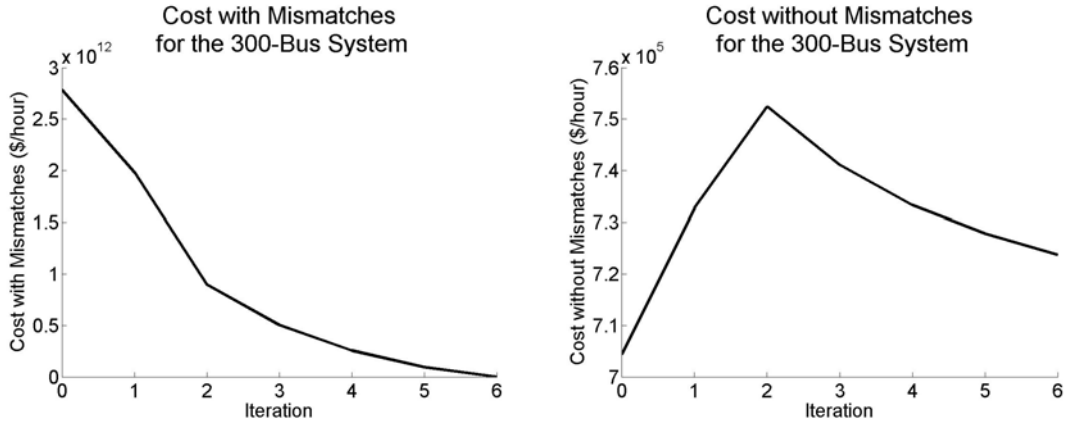


Figure 8.23 The costs with/without mismatches for the 300-bus system

8.6 Post-Solution Sensitivity Analysis

This section presents the shadow price in the final solution. Figure 8.24 shows the shadow price of $t_{109-111u} \leq t_{109-111u}^{\max}$, the upper bound of the control variable of the transformer between Bus 109 and Bus 111. Linear regression gives

$$Cost = 182676.73 - 73.09 \times t_{109-111u}^{\max} \text{ \$ / hour}, \quad (8-8)$$

so the shadow price $\frac{dCost}{dt_{109-111u}^{\max}} = -73.09 \text{ \$ / (hour} \cdot \text{pu)}$. That means the final cost reduces

when $t_{109-111u}^{\max}$ increases. Figure 8.25 shows the shadow price of $P_{g101,1}^{\min} \leq P_{g101,1}$, the real power variable of the first generator at Bus 101. Linear regression gives

$$Cost = 183476.18 + 8048.17P_{g101,1}^{\min} \$ / hour , \quad (8-9)$$

so the shadow price $\frac{dCost}{dP_{g101,1}^{\min}} = 8048.17 \$ / (hour \cdot pu)$. That means the final cost increases when $P_{g101,1}^{\min}$ increases. Therefore, using larger variable bounds means a lower final cost. In addition, a constraint with larger shadow price has more effect on the cost function. The data points of $t_{109-111u}^{\max}$ are not close to their linear regression since the shadow price of $t_{109-111u}^{\max}$ is very small and consequently the final cost is easily affected by linearization error.

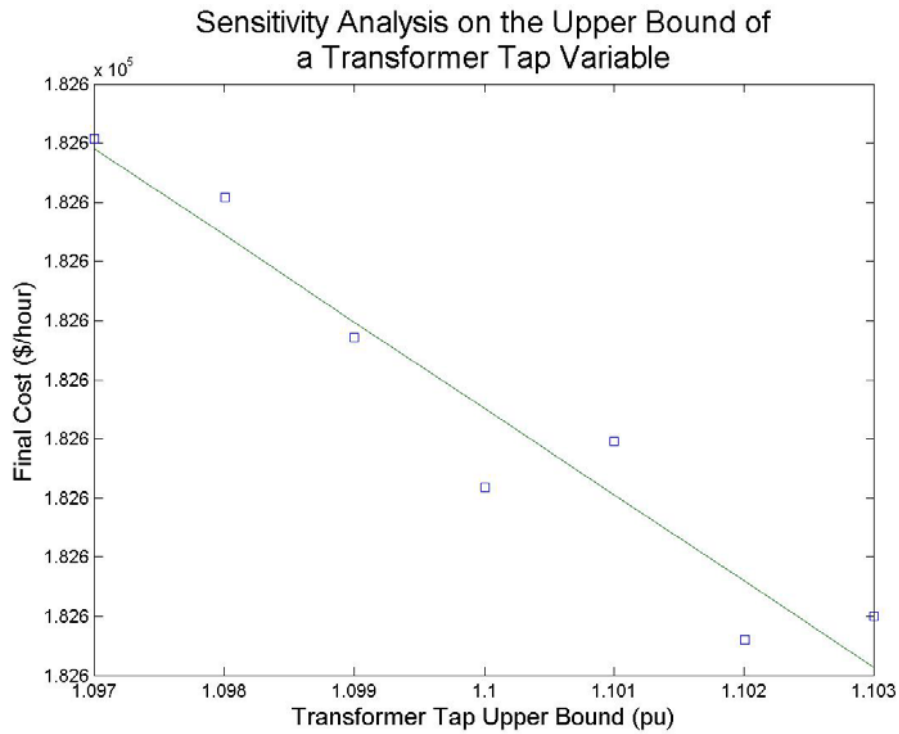


Figure8.24 Sensitivity analysis on the upper bound of a transformer-tap variable in the RTS-96 system

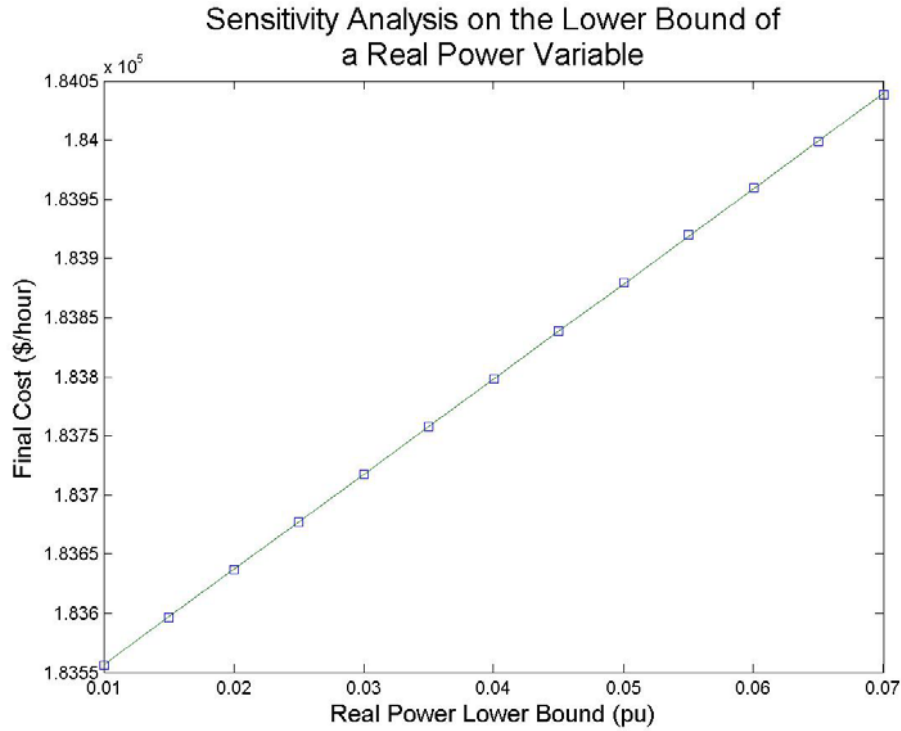


Figure 8.25 Sensitivity analysis on the lower bound of a real-power variable for the RTS-96 system

8.7 Summary

This chapter provided some illustrative examples to demonstrate the OPF algorithm developed in this thesis. First, a simple three-phase system was used for describing how to setup the nonlinear optimization problem. Next, the solution and the analysis of the RTS-79 system and the RTS-96 system were presented to elaborate on more properties of the algorithm. Then, the following section compared the proposed OPF algorithm with seven famous OPF software packages on nine widely used benchmark systems. Our OPF algorithm with parallelism can solve the 300-bus system in about 2 seconds. Finally, the post-solution sensitivity analysis was presented. This section gave the computation procedures of the shadow prices of two constraints, the upper bound of a transformer tap variable and the lower bound of a generator real power variable.

CHAPTER 9

DEMONSTRATION AND EVALUATION OF PROPOSED TOPF WITH SEVERAL TEST SYSTEMS

9.1 Introduction

This chapter demonstrates the proposed TOPF algorithm via three cases: an eight-bus system, the RTS-79 system, and the RTS-96 system. All these cases are modeled using quadratized three-phase power flow.

9.2 An Three-Phase Eight-Bus System Example

Figure 9.1 shows the eight-bus power system derived from the symmetric and balanced three-bus system in Section 8.2. The eight-bus system includes more types of devices for demonstration.

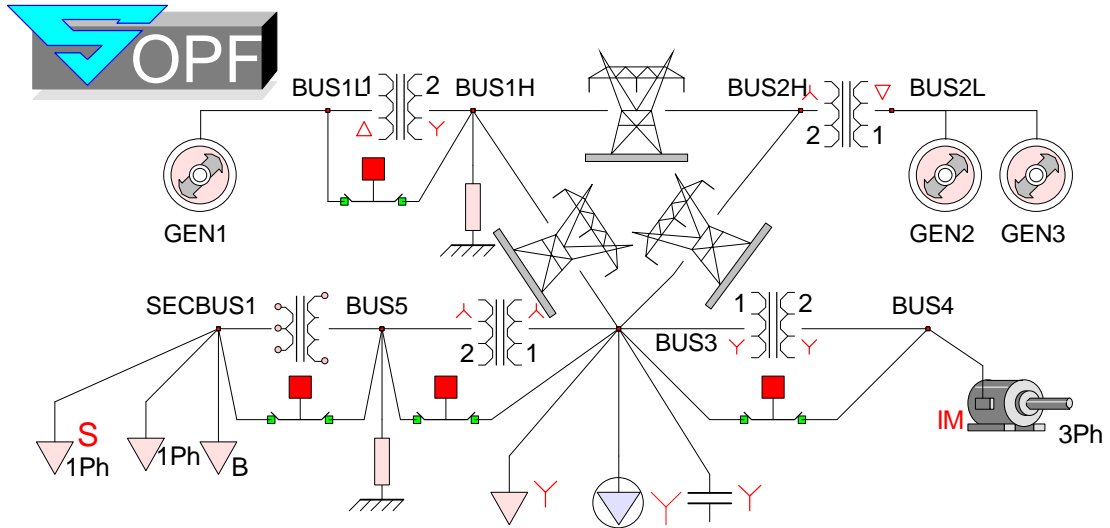


Figure 9.1 An eight-bus power system

Table 8-1 lists the generator parameters in the eight-bus system, as the three-bus system in Section 8.2, where GEN1 runs in slack mode, GEN2 runs in PV mode, and

GEN3 runs in PQ mode. The voltage level at the transmission level is $115kV$, the voltage level at the distribution level is $13.8kV$, and the voltage level at the residential level is $240V$. Figure 9.2 shows the parameters of the transmission line from BUS1H to BUS2H. The other two transmission lines have the same types of phase conductors, shields, and towers but different lengths: 15.0 miles (BUS1H to BUS3) and 5.0 miles (BUS2H to BUS3). The rating of the transformer connecting BUS2L and BUS 2H is $300MVA$, and the rating of other three-phase two-winding transformers are both $100MVA$. The power rating of the induction motor is $3MVA$ and its inertia constant is $0.2s$. Table 9-1 lists the parameters of all other loads.

3-Phase Overhead Transmission Line

Phase Conductors
 Type: ACSR
 Size: JOREE

Shields/Neutrals
 Type: HS
 Size: 5/16HS

Tower/Pole
 Type: 101A
 Circuit Number: 1
 Structure Name: N/A

Tower/Pole Ground Impedance (Ohms)
 R = 25.0 X = 0.0

Get From GS
 Line Length (miles): 20.0
 Line Span Length (miles): 0.1
 Soil Resistivity (Ohm-Meters): 100.0

Bus Name, Side 1: BUS1H
Circuit Number: 1
Bus Name, Side 2: BUS2H

Failure & Repair Rates
 Failure Rate (per year): 1.0
 Repair Rate (per year): 1.0

☐ Insulated Shields
☐ Transposed Phases
☐ Transposed Shields
 Read GPS Coordinates

Operating Voltage (kV): 115.0
Insulation Levels (kV)
 FOW (Front of Wave): 100.0
 BIL (Basic Insulation Level): 100.0
 AC (AC Withstand): 100.0

GA. Power H-Frame WoodPole TOWER

WinIGS-F - Form: IGSF_M102 - Copyright ? A. P. Meliopoulos 1998-2010

Figure 9.2 The transmission line (BUS1H to BUS2H) parameters in the eight-bus power system

Table 9-1: The load parameters in actual units in the eight-bus system

Device Type	Bus Name	Real Power	Reactive Power
Three-phase Constant Power Load	BUS3	200MW	58MVar
Three-phase Constant Impedance Load	BUS3	20MW	10MVar
Three-phase Capacitor Bank	BUS3	0MW	12MVar
Single-phase Constant Power Load	SECBUS1	10kW	3kVar
Single-phase Constant Impedance Load	SECBUS1	10kW	3kVar
Single-phase Balanced Constant Impedance Load	SECBUS1	5kW	3kVar

Since the three-phase model is much more complicated than the symmetric and balanced model, it has many more state variables. The formed TOPF problem has 123 state variables, where the induction motor has 30 state variables. The control variable set includes the capacitor bank switch, the voltage magnitude of GEN1, the real power and the voltage magnitude of GEN2, and the real power and the reactive power of GEN3. The penalty factor μ is set to 10^7 . The linearization bound is set to 0.2. Each phase at a three-phase bus is limited between $0.95pu$ and $1.05pu$. Single-phase bus voltages are limited between $0.90pu$ and $1.10pu$. Transformers are assumed to be non-controllable. Figure 9.3 shows the costs in each iteration. Table 9-2 shows detailed runtime information. The total runtime is 0.17 seconds. The total number of potential operating constraints is 72, while the algorithm adds 18 operating constraints in the end.

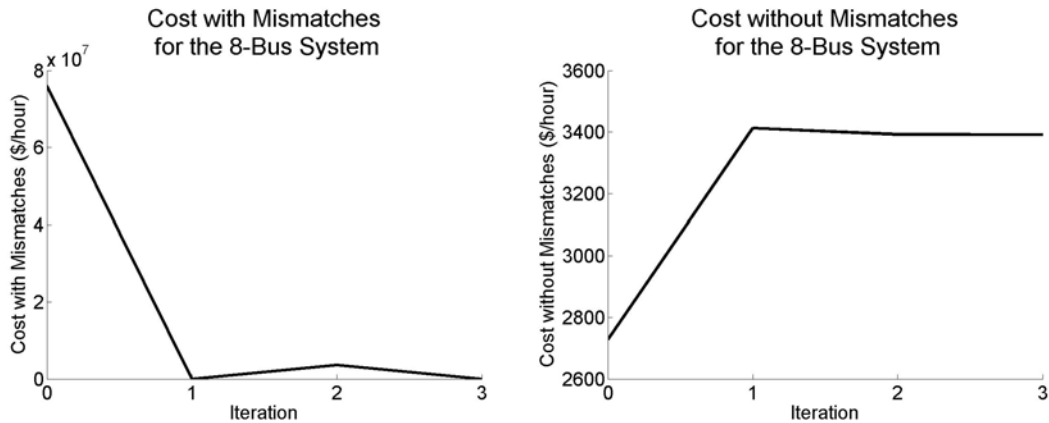
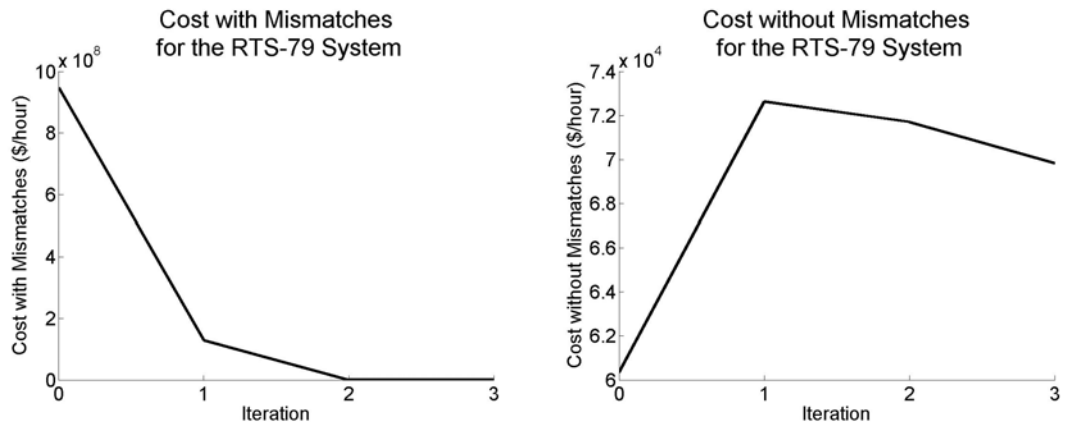
**Figure 9.3 The costs with/without mismatches for the eight-bus system**

Table 9-2: Runtime information of the eight-bus system

#	Layer-3 Runtime	Linearization Runtime	LP Runtime	Power Flow Runtime	# Constraint
1-1-1	0.028843	0.000411	0.000621	0.028777	2
1	0.028861				
2-1-1	0.029817	0.000407	0.000136	0.027756	18
2-2-1	0.056166	0.001418	0.000242	0.027231	18
2	0.086027				
3-1-1	0.030158	0.001667	0.000247	0.026749	18
3	0.030173				

9.3 The Three-Phase RTS-79 System Example

Figure 9.5 shows the three-phase RTS-79 system derived from the symmetric and balanced RTS-79 system in Section 8.3 [90]. The penalty factor μ is set to 10^7 . The linearization bound is set to 6.68×10^{-7} . The bus voltages are limited between $0.95pu$ and $1.05pu$. The taps of the transformers connecting substations are between $0.9pu$ and $1.1pu$, and step-up transformers are non-controllable. The TOPF problem has 68 control variables and 786 state variables. Figure 9.4 shows the costs in each iteration. Table 9-3 shows the detailed runtime information. The total runtime is 5.37 seconds. The total number of the operating constraints is 539, while there are 152 active operating constraints in the last iteration.

**Figure 9.4 The costs with/without mismatches for the three-phase RTS-79 system**

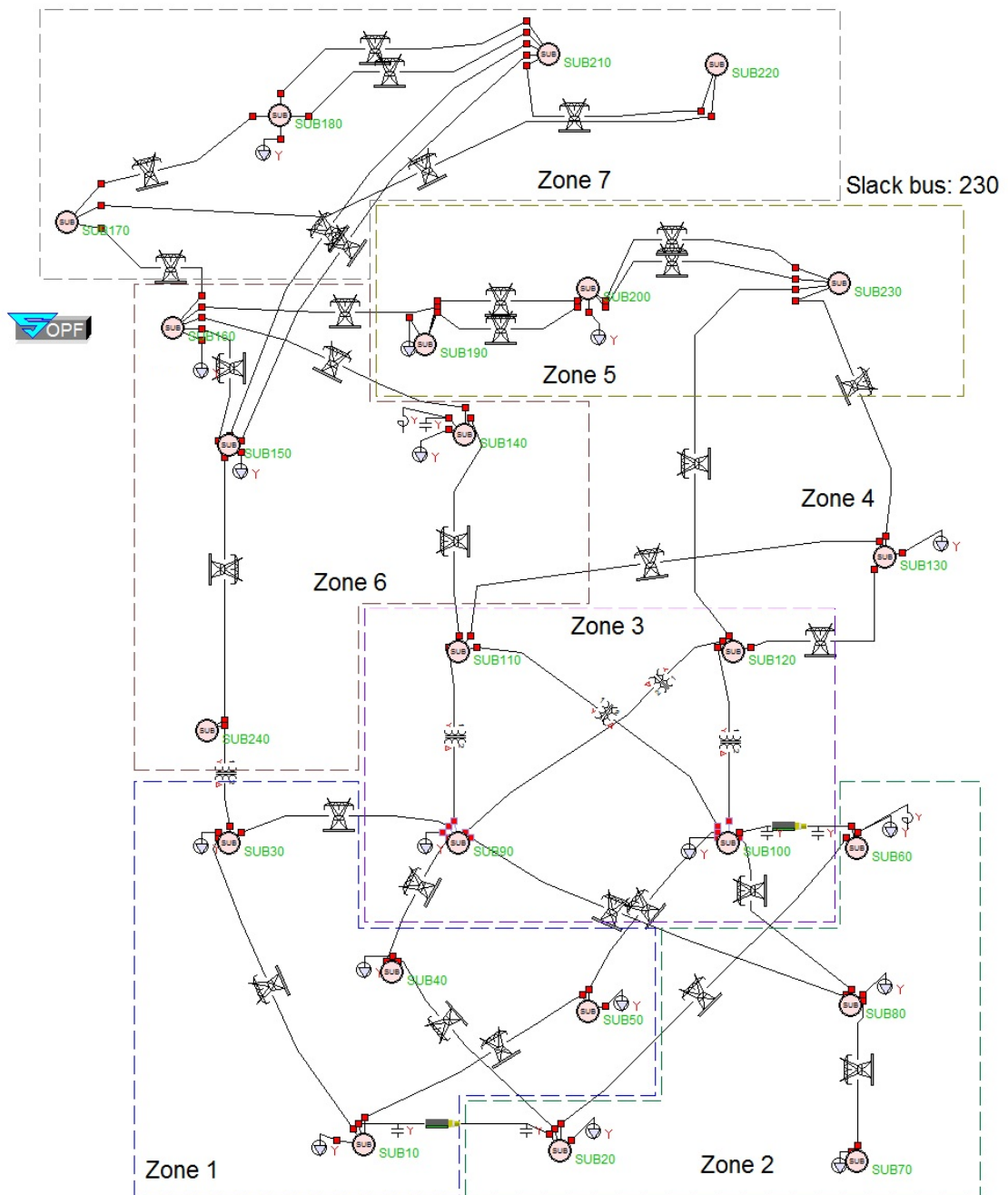


Figure 9.5 The three-phase RTS-79 power system

Table 9-3: Runtime information of the three-phase RTS-79 system

#	Layer-3 Runtime	Linearization Runtime	LP Runtime	Power Flow Runtime	# Constraint
1-1-1	0.173266	0.007988	0.000407	0.167586	2
1-2-1	0.630574	0.270877	0.006850	0.196492	124
1-2-2	0.337500	0.000426	0.008578	0.167485	124
1-2-3	0.329287	0.000403	0.008001	0.166152	124
1-2-4	0.330533	0.000601	0.007845	0.166279	124
1-2-5	0.329061	0.000425	0.007510	0.166624	124
1-2-6	0.333342	0.000424	0.007524	0.168047	124
1	2.463710				
2-1-1	0.460630	0.271801	0.005636	0.166114	124
2-1-2	0.341018	0.000404	0.005856	0.168388	124
2-1-3	0.339616	0.000403	0.005962	0.167369	152
2-2-1	0.405417	0.063222	0.010263	0.167295	152
2-2-2	0.341846	0.000494	0.008652	0.167667	152
2	1.888770				
3-1-1	0.526419	0.334408	0.010106	0.165554	152
3-1-2	0.343638	0.000677	0.011355	0.167005	152
3	0.870164				

9.4 The Three-Phase RTS-96 System Example

Figure 9.6 presents the sketch map of the RTS-96 system derived from the three-phase RTS-79 system in Section 9.3. The bold lines show the interconnections between the substations in different areas. Area 3 has one more controllable transformer compared with the RTS-79 system.

The linearization bound is set to 2.933×10^{-7} and the other parameters are the same as the RTS-79 system. The formed TOPF problem has 205 control variables and 2,360 state variables. Figure 9.7 shows the costs in each iteration. Table 9-4 shows detailed runtime information. The total runtime is 43.36 seconds. The RTS-96 system has 1,631 operating constraints, and the algorithm adds 275 active operating constraints in the end.

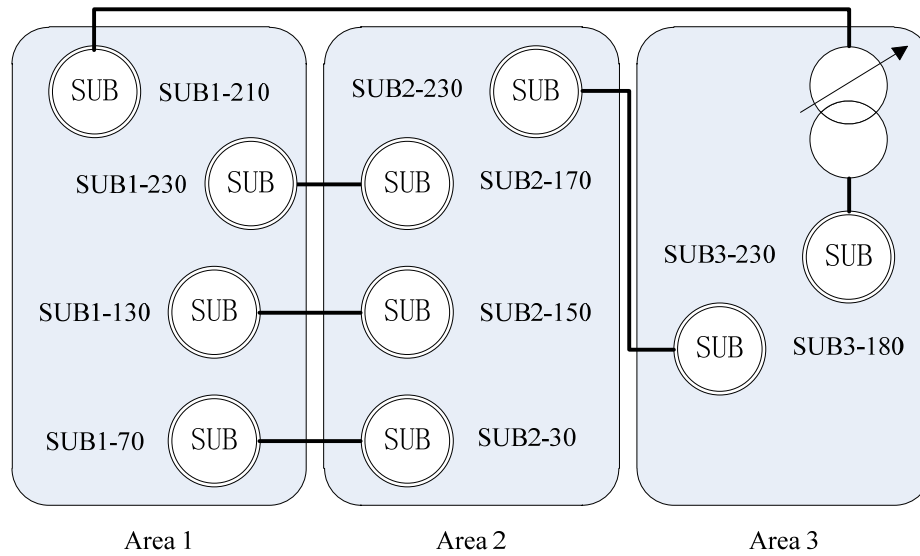


Figure 9.6 The sketch map of the three-phase RTS-96 power system

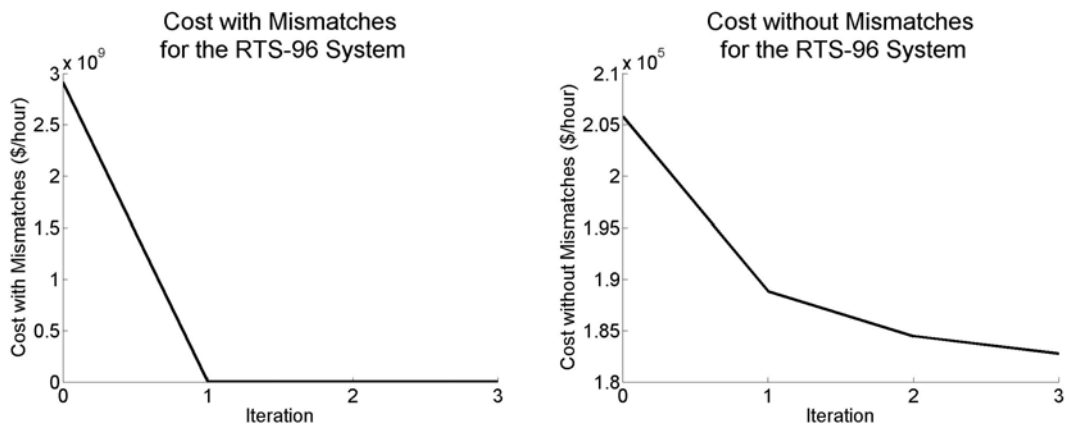


Figure 9.7 The costs with/without mismatches for the three-phase RTS-96 system

Table 9-4: Runtime information of the three-phase RTS-79 system

#	Layer-3 Runtime	Linearization Runtime	LP Runtime	Power Flow Runtime	# Constraint
1-1-1	0.86983	0.05896	0.00117	0.73966	2
1-2-1	5.61849	4.33786	0.05137	0.67394	228
1-2-2	1.29183	0.00242	0.06247	0.67276	228
1-2-3	1.27127	0.00239	0.05865	0.65314	228
1-2-4	1.27638	0.00239	0.06329	0.65599	228
1-2-5	1.27033	0.00226	0.05823	0.66007	228
1-2-6	1.25060	0.00231	0.06020	0.63686	228
1-2-7	1.25575	0.00238	0.05679	0.64298	238
1-3-1	1.47544	0.19776	0.06647	0.64920	238
1-3-2	1.26452	0.00249	0.06662	0.64260	238
1-3-3	1.26754	0.00236	0.06606	0.64074	238
1-3-4	1.27787	0.00249	0.06747	0.65006	238
1	19.3903				
2-1-1	5.41201	4.56495	0.05242	0.68547	238
2-1-2	1.35438	0.00238	0.05597	0.64028	238
2-1-3	1.35409	0.00239	0.05983	0.64367	270
2-2-1	2.04234	0.63359	0.07140	0.67570	270
2-2-2	1.40742	0.00269	0.07469	0.66522	270
2	11.5709				
3-1-1	6.07620	5.23366	0.07252	0.67042	270
3-1-2	1.43808	0.00280	0.07180	0.68559	270
3-1-3	1.39747	0.00288	0.07070	0.65693	275
3-2-1	1.53588	0.10936	0.07759	0.67652	275
3-2-2	1.41834	0.00301	0.07692	0.67573	275
3	11.8663				

9.5 Summary

This chapter presented the TOPF solution of three test systems: the eight-bus system, the RTS 79 system, and the RTS-96 system. All three systems are modeled via the quadratized three-phase power system model. The results showed that the proposed TOPF algorithm is effective for small- and middle- size systems. Tests for large systems are left for future work.

CHAPTER 10

DEMONSTRATION OF REACTIVE SOURCE PLANNING WITH DYNAMIC PROGRAMMING AND THE PROPOSED OPF METHOD

10.1 Introduction

This chapter presents a base system that may already have a number of static and dynamic VAR resources. The operation of the system is over a planning horizon, typically five to fifteen years is considered. Over this planning period it may be necessary to add dynamic VAR sources and/or static VAR sources to maintain acceptable performance as electric loads increase. The dynamic sources can be of various types, such as synchronous generators, STATCOMs, static VAR compensators, inverter based interfaces of wind, PV systems, etc. The decision process involves the addition of specific static and dynamic VAR sources at specific locations in the system at specific times (stages). Then, in terms of the installed sources, the state of the system can be defined at a given time as the base case plus the addition of specific amount of static and dynamic VAR sources to specific locations. In general, it is assumed that a decision to add VAR sources can be taken at specific time intervals (or stages), for example at intervals of six months. In this case, a stage is equivalent to a period of six months. In a planning period of ten years, there will be a total of twenty stages.

10.2 System Description

The test system in Figure 10.1 consists of transmission, two substations, and distribution feeders. Substations are shown in Figures 10.2 and Figure 10.3. The cost coefficients of the 155MVA generators are $a = 382.2391$ \$/hour, $b = 12.38826$ \$/(MW·hour), and $c = 0.008342$ \$/(MW²·hour). The cost coefficients of the 350MVA

generators are $a = 665.1094$ $\$/hour$, $b = 11.84954$ $\$/(MW \cdot hour)$, and $c = 0.004895$ $\$/(MW^2 \cdot hour)$. Each of the two substations includes a $15MVA$, $115kV/13.8kV$ transformer with three distribution feeders comprising 24 $13.8kV$ buses, 11 induction motors, 6 three-phase loads, and 13 single-phase loads.

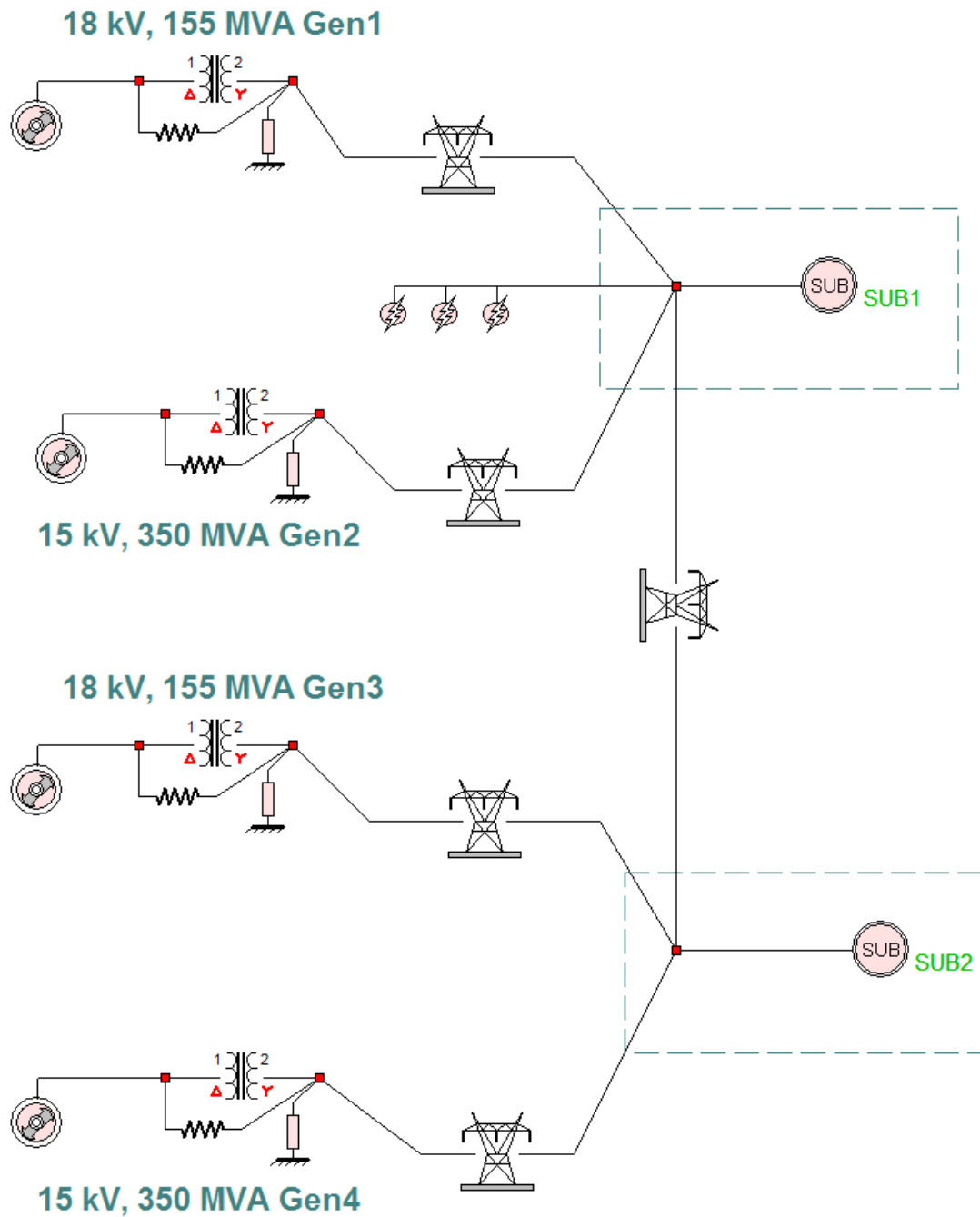


Figure 10.1 The test system for dynamic VAR planning

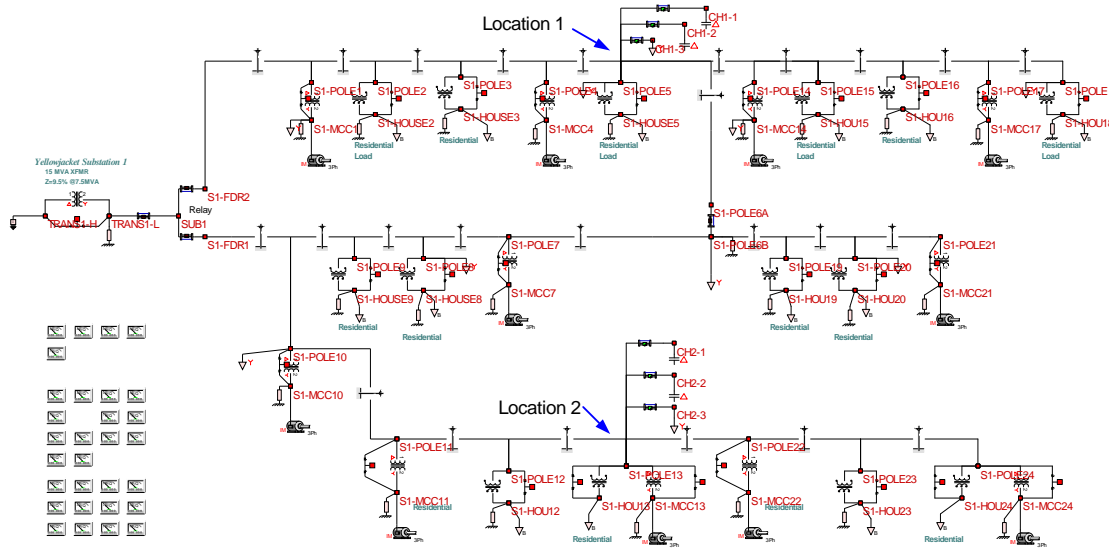


Figure 10.2 Substation 1 in the test system for dynamic VAR planning

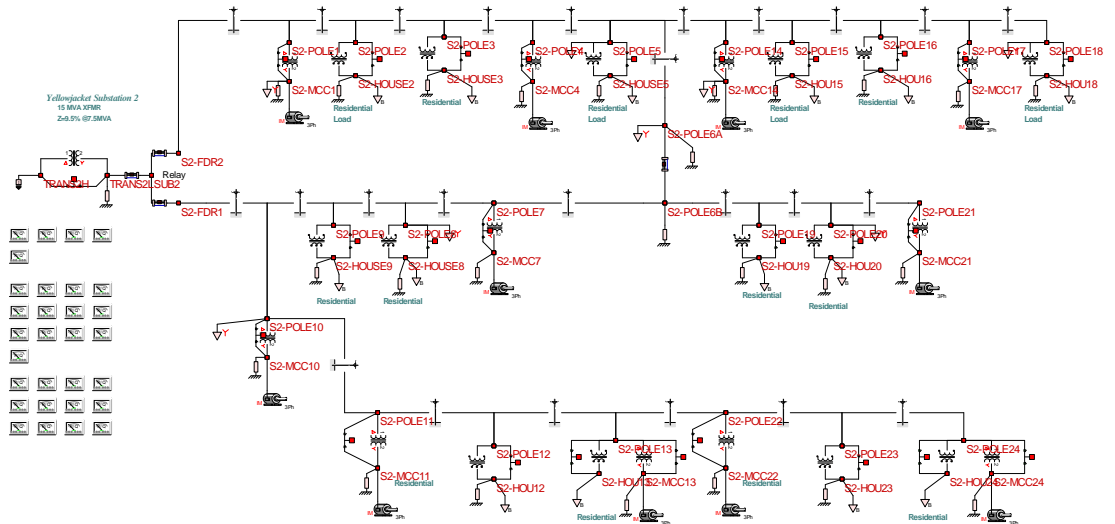


Figure 10.3 Substation 2 in the test system for dynamic VAR planning

A three-phase fault was considered at the high voltage side of the transformer at Substation 1. This type of faults at this location usually causes the most severe effect on induction motors. Candidate locations for reactive power allocation are marked in Figure 10.2 and Figure 10.3. The amount of reactive power allocated at each location is 900 *kVar* and 1,500 *kVar* (for static sources) and 600 *kVar* (for dynamic sources). Table 10-1 shows the different options for reactive power allocation. The transitions between these

states are considered over a period of twenty planning stages. Each stage is separated by a period of six months. It has also been assumed that a reactive power source cannot be removed once it has been installed.

10.3 Candidate Reactive Sources

Table 10-1 shows different reactive sources considered. The acquisition cost and the installation cost are considered for each type of device, and they will be converted to the annualized equivalent cost (AEC).

Table 10-1: Candidate reactive sources

#	Device Type	Capacity (kVar)	Acquisition Cost (\$)	Installation Cost (\$)	Annualized Equivalent Cost (\$)
0	Capacitor Bank	900	8,500	5,500	840.18
1	Capacitor Bank	1500	9,500	6,500	960.21
2	Static VAR Compensator	600	20,000	7,000	1620.4

10.4 Candidate Locations Selection

Table 10-2 shows the locations to place the capacitor banks or the SVCs. Candidate locations are selected from these locations with larger negative sensitivities and certain distances between each other. Therefore, S1-POLE5 and S1-POLE13 are selected, where S1-POLE5 is at Location 1 and S1-POLE13 is at Location 2 in Figure 10.2 respectively.

Table 10-2: Locations of capacitor bank with sensitivity calculation

#	Location Name	Sensitivity	#	Location Name	Sensitivity
0	S1-POLE1	6.54895	24	S2-POLE2	29.4894
1	S1-POLE2	7.33222	25	S2-POLE3	21.6163
2	S1-POLE3	-8.0637	26	S2-POLE4	9.30731
3	S1-POLE4	8.37319	27	S2-POLE5	28.8481
4	S1-POLE5	-26.3786	28	S2-POLE14	-5.52452
5	S1-POLE14	10.0024	29	S2-POLE15	30.1988
6	S1-POLE15	9.81778	30	S2-POLE16	29.2361
7	S1-POLE16	9.98795	31	S2-POLE17	29.6444
8	S1-POLE17	10.6499	32	S2-POLE18	30.8978
9	S1-POLE18	7.879	33	S2-POLE21	27.768
10	S1-POLE9	5.88635	34	S2-POLE20	25.2289
11	S1-POLE8	-27.2261	35	S2-POLE19	-6.36113
12	S1-POLE7	13.2733	36	S2-POLE7	32.1202
13	S1-POLE10	-29.1543	37	S2-POLE8	-6.97761
14	S1-POLE11	-31.9043	38	S2-POLE9	-7.31361
15	S1-POLE12	8.1601	39	S2-POLE10	26.0613
16	S1-POLE13	-33.0314	40	S2-POLE11	-3.05085
17	S1-POLE19	14.1361	41	S2-POLE12	24.3671
18	S1-POLE20	13.1129	42	S2-POLE13	21.9133
19	S1-POLE21	13.7967	43	S2-POLE22	23.7267
20	S1-POLE22	8.50671	44	S2-POLE23	24.1174
21	S1-POLE23	8.45695	45	S2-POLE24	25.6656
22	S1-POLE24	8.62304	46	S1-POLE6B	2.58117
23	S2-POLE1	-5.7388	47	S2-POLE6B	-7.07904

10.5 State Definitions

This section describes the state definitions. Each state at any stage is differentiated according to different reactive source allocations. By considering possible combinations of these choices, there are 16 states in total shown in Table 10-3, including the base case that does not contain any addition of the reactive power resources.

Two candidate allocation choices are selected. Each state is associated with a certain cost since it involves a certain addition of reactive power sources. Five types of costs are associated to each state: AEC $J_{AnnualizedEquivalentCost}$, the operating cost $J_{OperatingCost}$, the voltage deviation penalty $J_{VoltageDeviationPenalty}$, the voltage recovery time penalty $J_{VoltageRecoveryTimePenalty}$, the voltage oscillation penalty $J_{VoltageOscillationPenalty}$, the lower bound of voltage magnitudes $J_{VoltageLowerBound}$, and the upper bound of voltage recovery times $J_{VoltageRecoveryTimeUpperBound}$. AEC is derived from the acquisition cost and the installation cost. It occurs when a new reactive resource is installed and remains the same in

subsequent stages. The operating cost is calculated from the TOPF result. The voltage deviation penalty, the voltage recovery time penalty, and the voltage oscillation penalty occur in every planning stage recurrently. They are different in various stages since the system parameters may change during stages.

Table 10-3 shows the state definitions with the acquisition cost, the installation cost, and the AEC since their values are the same for the states with the same index in different stages. For example, these costs are the same at State 3, Stage 2 and Stage 3, Stage 3, respectively. However, the operating cost and the performance penalties vary in different stages.

Table 10-3: State definitions

State #	Location	Capacity (kVar)	Type	Acquisition Cost (\$)	Installation Cost (\$)	Annualized Equivalent Cost (\$)
0	Base System					
1	CH2-1	900	Capacitor Bank	8500	5500	840.182
2	CH2-2	1500	Capacitor Bank	9500	6500	960.208
3	CH2-3	600	Static VAR Compensator	20000	7000	1620.35
4	CH1-1	900	Capacitor Bank	8500	5500	840.182
5	CH1-1	900	Capacitor Bank	17000	11000	1680.36
	CH2-1	900	Capacitor Bank			
6	CH1-1	900	Capacitor Bank	18000	12000	1800.39
	CH2-2	1500	Capacitor Bank			
7	CH1-1	900	Capacitor Bank	28500	12500	2460.53
	CH2-3	600	Static VAR Compensator			
8	CH1-2	1500	Capacitor Bank	9500	6500	960.208
9	CH1-2	1500	Capacitor Bank	18000	12000	1800.39
	CH2-1	900	Capacitor Bank			
10	CH1-2	1500	Capacitor Bank	19000	13000	1920.42
	CH2-2	1500	Capacitor Bank			
11	CH1-2	1500	Capacitor Bank	29500	13500	2580.56
	CH2-3	600	Static VAR Compensator			
12	CH1-3	600	Static VAR Compensator	20000	7000	1620.35
13	CH2-1	900	Capacitor Bank	28500	12500	2460.53
	CH1-3	600	Static VAR Compensator			
14	CH2-2	1500	Capacitor Bank	29500	13500	2580.56
	CH1-3	600	Static VAR Compensator			
15	CH1-3	600	Static VAR Compensator	40000	14000	3240.7
	CH2-3	600	Static VAR Compensator			

10.6 Simulation Results

We simulate every state in all the stages. The simulation time for each case is 3 seconds. A three-phase fault occurs at 0.1s and clears at 0.3s. Table 10-4 and Table 10-5

show the simulation result of State 6 (one capacitor at each location) at Stage 19 for all the load buses. Since the location of the three-phase fault is closer to Substation 1, the voltage recovery times of the buses in Substation 1 is longer.

Table 10-4: Simulation data of Substation 1 for State 6 at Stage 19

Load Bus Name	Nominal Voltage (kV)	Actual Voltage (kV)	Voltage Deviation Percentage (%)	Recovery Time (s)	Voltage Oscillation Magnitude (kV)	Voltage Oscillation Percentage (%)
STAGE - 19, STATE - 6, SUBSTATION - 1						
S1-POLE5_A	7.96743	8.51567	6.88094	0.435	0.0960342	1.20533
S1-POLE6B_A	7.96743	8.51143	6.82778	0.435	0.0959873	1.20475
S1-POLE8_A	7.96743	8.48735	6.5255	0.44	0.0958447	1.20296
S1-POLE10_A	7.96743	8.34538	4.74358	0.455	0.0941182	1.18129
S1-POLE18_A	7.96743	8.55424	7.36511	0.43	0.0968623	1.21573
S1-MCC1_A	0.277128	0.297674	7.41379	0.41	0.00336017	1.2125
S1-MCC4_A	0.277128	0.297144	7.22251	0.435	0.00340506	1.2287
S1-MCC7_A	0.277128	0.292781	5.64811	0.475	0.00340491	1.22864
S1-MCC10_A	0.277128	0.290744	4.91316	0.46	0.00334894	1.20844
S1-MCC11_A	0.277128	0.286981	3.55538	0.52	0.00336835	1.21545
S1-MCC13_A	0.277128	0.283271	2.21678	1.09	0.0063457	2.28981
S1-MCC14_A	0.277128	0.300748	8.52309	0.425	0.0034465	1.24365
S1-MCC17_A	0.277128	0.300079	8.28181	0.43	0.00344339	1.24253
S1-MCC21_A	0.277128	0.299809	8.18424	0.48	0.0035825	1.29272
S1-MCC22_A	0.277128	0.287754	3.83427	0.52	0.00338141	1.22016
S1-MCC24_A	0.277128	0.287492	3.73982	0.52	0.00338278	1.22066
S1-HOUSE2_L1	0.12	0.126727	5.60601	0.44	0.00142527	1.18773
S1-HOUSE3_L1	0.12	0.131538	9.6147	0.41	0.00147241	1.22701
S1-HOUSE5_L1	0.12	0.12992	8.26636	0.425	0.0014883	1.24025
S1-HOUSE8_L1	0.12	0.131423	9.51928	0.415	0.00147516	1.2293
S1-HOUSE9_L1	0.12	0.126242	5.20203	0.45	0.00142563	1.18802
S1-HOU12_L1	0.12	0.122788	2.32306	0.66	0.00142622	1.18852
S1-HOU13_L1	0.12	0.122735	2.27917	1.05	0.00142749	1.18958
S1-HOUSE15_L1	0.12	0.128131	6.77545	0.435	0.00145087	1.20906
S1-HOUSE16_L1	0.12	0.133034	10.8614	0.405	0.00149731	1.24776
S1-HOUSE18_L1	0.12	0.130724	8.93668	0.42	0.00150551	1.25459
S1-HOUSE19_A	0.12	0.126049	5.04044	0.46	0.00142536	1.1878
S1-HOUSE20_A	0.12	0.1328	10.6664	0.415	0.00149063	1.24219
S1-HOUSE23_L1	0.12	0.122664	2.22013	1.05	0.00142479	1.18732
S1-HOUSE24_L1	0.12	0.122703	2.25286	1.05	0.00142712	1.18926

Table 10-5: Simulation data of Substation 2 for State 6 at Stage 19

Load Bus Name	Nominal Voltage (kV)	Actual Voltage (kV)	Voltage Deviation Percentage (%)	Recovery Time (s)	Voltage Oscillation Magnitude (kV)	Voltage Oscillation Percentage (%)
STAGE - 19, STATE - 6, SUBSTATION - 2						
S2-POLE5_A	7.96743	8.49739	6.65151	0.265	0.103191	1.29516
S2-POLE6B_A	7.96743	8.49506	6.62223	0.265	0.103176	1.29497
S2-POLE8_A	7.96743	8.46484	6.24295	0.265	0.102228	1.28307
S2-POLE10_A	7.96743	8.32517	4.48992	0.27	0.0923662	1.1593
S2-POLE18_A	7.96743	8.52733	7.02732	0.265	0.105656	1.3261
S2-MCC1_A	0.277128	0.296485	6.98491	0.27	0.00526517	1.8999
S2-MCC4_A	0.277128	0.296139	6.8599	0.275	0.00436169	1.57389
S2-MCC7_A	0.277128	0.295323	6.56552	0.285	0.00467892	1.68836
S2-MCC10_A	0.277128	0.291188	5.07339	0.27	0.00349619	1.26158
S2-MCC11_A	0.277128	0.283383	2.25701	0.285	0.00314761	1.1358
S2-MCC13_A	0.277128	0.283957	2.46417	0.285	0.00315329	1.13785
S2-MCC14_A	0.277128	0.298794	7.81783	0.265	0.00370784	1.33795
S2-MCC17_A	0.277128	0.298577	7.73958	0.27	0.00404184	1.45847
S2-MCC21_A	0.277128	0.299015	7.89786	0.27	0.00405815	1.46436
S2-MCC22_A	0.277128	0.285511	3.02491	0.285	0.00316336	1.14148
S2-MCC24_A	0.277128	0.285967	3.18946	0.29	0.00316724	1.14288
S2-HOUSE2_L1	0.12	0.126468	5.3896	0.27	0.00149357	1.24464
S2-HOUSE3_L1	0.12	0.130686	8.90475	0.255	0.00144244	1.20203
S2-HOUSE5_L1	0.12	0.129355	7.79608	0.265	0.00158085	1.31737
S2-HOUSE8_L1	0.12	0.130491	8.74285	0.255	0.001441	1.20083
S2-HOUSE9_L1	0.12	0.125922	4.93498	0.27	0.00148119	1.23433
S2-HOU12_L1	0.12	0.121939	1.61559	0.285	0.00134106	1.11755
S2-HOU13_L1	0.12	0.12186	1.55011	0.285	0.00134067	1.11722
S2-HOUSE15_L1	0.12	0.126905	5.75453	0.27	0.0015377	1.28142
S2-HOUSE16_L1	0.12	0.131428	9.52307	0.26	0.00148682	1.23902
S2-HOUSE18_L1	0.12	0.129809	8.1743	0.265	0.00161846	1.34872
S2-HOUSE19_A	0.12	0.127667	6.3888	0.27	0.0015765	1.31375
S2-HOUSE20_A	0.12	0.131907	9.92263	0.26	0.00148699	1.23916
S2-HOUSE23_L1	0.12	0.121747	1.45562	0.29	0.00133971	1.11643
S2-HOUSE24_L1	0.12	0.121744	1.45316	0.29	0.0013398	1.1165

10.7 Cost Evaluation

This section presents the cost computation and gives a computational example for State 6 at Stage 19.

10.7.1 Transition costs between Stages

The transition costs are summarized in Table 10-6. Each row represents a state in the present stage and each column represents a state in the next stage. ∞ means the transition is not feasible since the removal of any installed reactive source is not allowed.

Table 10-6: Transition Costs, Stage $k-1$ to Stage k

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1		1	∞	∞	∞	1	∞	∞	∞	1	∞	∞	∞	1	∞	∞
2			1	∞	∞	∞	1	∞	∞	∞	1	∞	∞	∞	1	∞
3				1	∞	∞	∞	1	∞	∞	∞	1	∞	∞	∞	1
4					1	1	1	1	∞	∞	∞	∞	∞	∞	∞	∞
5						1	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
6							1	∞	∞	∞	∞	∞	∞	∞	∞	∞
7								1	∞	∞	∞	∞	∞	∞	∞	∞
8									1	1	1	1	∞	∞	∞	∞
9										1	∞	∞	∞	∞	∞	∞
10											1	∞	∞	∞	∞	∞
11												1	∞	∞	∞	∞
12													1	1	1	1
13														1	∞	∞
14															1	∞
15																1

10.7.2 Operating Costs computed via TOPF

The planning problem has 321 states in total. Stage 0 has only one state since no additional VAR sources exist at the beginning. TOPF should be run for on all these states and obtain the operating costs for them. The penalty factor is set to 1×10^7 and the linearization limit is set to 1.1×10^{-6} . The total runtime of all states is 1,933.06s. Therefore, the average runtime for each case is 6.02s, where the maximum runtime is 9.90s and the minimum runtime is 4.91s. All these cases converge within five iterations. The average number of operating constraints added is 12.32 (maximum = 25 and minimum = 8). The detailed information of runtime and operating constraints is not provided for concision.

Table 10-7 lists TOPF results from Stage 0 to Stage 20. The unit of these numbers are \$/hour, so the operating cost in each planning stage (= 0.5 year) equals its corresponding TOPF result times 4,380 hours (= $365 \times 24/2$). For example, the operating cost at Stage 19, State 6 = $2,637.8053 \times 4,380$.

Table 10-7: TOPF results from Stage 0 to Stage 20

Stage\State	0	1	2	3	4	5	6	7
0	2503.7335							
1	2509.3920	2513.4198	2515.8172	2512.2894	2513.4116	2516.9063	2519.8144	2515.7324
2	2515.3028	2519.2053	2521.6450	2517.2813	2519.2332	2522.7606	2525.6851	2521.5442
3	2521.9845	2525.0762	2527.5569	2522.9995	2525.0962	2528.6943	2531.2592	2527.4611
4	2528.7020	2530.1153	2533.5512	2528.9167	2530.1249	2534.7102	2537.3325	2533.4605
5	2535.6335	2536.1634	2539.6129	2534.9417	2536.1785	2540.8087	2543.4455	2539.541
6	2542.5884	2542.2966	2545.0015	2541.0571	2542.3174	2546.9874	2549.6651	2544.7956
7	2549.7792	2548.5154	2551.1802	2547.2579	2548.5423	2552.4886	2555.9705	2551.0569
8	2557.0974	2554.8205	2557.6817	2553.7496	2555.0558	2558.7336	2562.3803	2557.4051
9	2564.5692	2561.2123	2563.9005	2560.6109	2561.7567	2565.1879	2567.9867	2563.8405
10	2572.1984	2568.5327	2570.4169	2568.2278	2569.3985	2571.7301	2574.5674	2570.3636
11	2579.9918	2574.4425	2577.0212	2575.8595	2577.0589	2578.3606	2581.2369	2576.975
12	2587.9520	2581.8338	2583.7136	2583.7613	2584.9394	2585.0798	2587.9951	2584.6399
13	2596.0774	2589.9439	2590.4945	2591.7396	2592.9378	2591.8878	2594.8426	2590.4633
14	2604.3731	2597.9955	2597.3641	2599.9389	2601.1571	2600.0623	2602.7932	2597.4769
15	2612.8709	2606.3164	2604.3224	2608.3007	2609.5411	2605.7713	2610.0454	2605.2769
16	2621.5710	2614.8066	2613.2843	2616.8392	2618.1027	2612.8469	2615.9217	2613.7569
17	2645.6468	2623.4886	2620.6589	2625.5773	2626.8486	2620.5580	2623.127	2622.4148
18	2655.2712	2632.3551	2628.2477	2634.523	2635.8131	2629.1871	2630.4217	2631.2108
19	2665.1168	2641.4404	2655.0054	2643.6763	2660.0466	2656.7177	2637.8053	2655.032
20	2674.2545	2666.1065	2662.8397	2668.5409	2669.9683	2664.5836	2668.1132	2664.7218
Stage\State	8	9	10	11	12	13	14	15
0								
1	2515.7592	2519.1151	2522.3089	2518.5073	2511.4261	2515.7276	2518.596	2514.5411
2	2521.5742	2525.6147	2528.2176	2523.9709	2518.0724	2521.5788	2524.0558	2520.3791
3	2527.4945	2531.1916	2534.2318	2529.9254	2523.0159	2527.4735	2529.9911	2526.2567
4	2533.5183	2537.2466	2540.3404	2535.9617	2528.9364	2533.4713	2536.0359	2532.2594
5	2539.5816	2543.3851	2546.0957	2542.0819	2535.1630	2539.5500	2542.1507	2537.3734
6	2545.7681	2549.6089	2552.3568	2548.2873	2541.2916	2544.8022	2548.3502	2543.5284
7	2551.1067	2555.9185	2558.7006	2553.6916	2547.9083	2551.1253	2553.7473	2549.8792
8	2557.5887	2562.3318	2565.1324	2560.0817	2555.2412	2557.4075	2560.1309	2556.0968
9	2563.8992	2567.9441	2571.6509	2566.5595	2562.6299	2563.8407	2566.6020	2562.5114
10	2570.4270	2574.5296	2578.3132	2573.1256	2570.2658	2570.3616	2573.1611	2569.0134
11	2577.0433	2581.2039	2584.2085	2579.7804	2577.9561	2576.9705	2579.8086	2576.4948
12	2583.9150	2588.5033	2590.9616	2586.5241	2585.8735	2583.6678	2586.5447	2582.2815
13	2591.0381	2594.8201	2597.9306	2593.3571	2593.9402	2590.4538	2593.3697	2589.7997
14	2599.0893	2601.7626	2604.8376	2601.3797	2602.1814	2597.3287	2600.2840	2597.9362
15	2607.5264	2608.7947	2611.9105	2607.2916	2610.5944	2604.3950	2607.2874	2606.2816
16	2616.0102	2615.9166	2619.0734	2614.3932	2619.1849	2612.7647	2614.3801	2614.7498
17	2624.6836	2623.1282	2626.3262	2621.5844	2627.9913	2621.5311	2621.5620	2623.3848
18	2633.5532	2632.3314	2633.6688	2630.8985	2637.0038	2630.2217	2630.9353	2632.2392
19	2657.5213	2639.9693	2641.1009	2658.4942	2646.2243	2655.0016	2636.4076	2656.0835
20	2667.3606	2668.1387	2671.7588	2666.3884	2671.3813	2663.4841	2666.3285	2665.9279

10.7.3 Costs at Stage 19

Table 10-8 shows the costs of each state at Stage 19. Several bus voltages cannot recover after the fault clears, so the addition of VAR sources is necessary.

Table 10-8: Cost data for all states at Stage 19

State #	Annualized Equivalent Cost	Operating Cost	Voltage Deviation Penalty	Voltage Recovery Time Penalty	Voltage Oscillation Penalty	Voltage Lower Bound	Voltage Recovery Time Upper Bound	State Cost
0	0	11,673,211.58	120.68	∞	0	0	∞	∞
1	840.18	11,569,508.95	142.03	∞	0	0	∞	∞
2	960.21	11,628,923.65	161.14	454.2	1.55	0	0	11,630,500.74
3	1,620.35	11,579,302.19	133.92	∞	0	0	∞	∞
4	840.18	11,651,004.11	147.3	∞	0	0	∞	∞
5	1,680.36	11,636,423.53	177.2	355.5	1.64	0	0	11,638,638.23
6	1,800.39	11,553,587.21	202.28	150.47	1.62	0	0	11,555,741.98
7	2,460.53	11,629,040.16	166.08	1,047.72	1.43	0	0	11,632,715.93
8	960.21	11,639,943.29	170.1	2,380.62	309.94	0	∞	∞
9	1,800.39	11,563,065.53	205.97	174.47	1.67	0	0	11,565,248.03
10	1,920.42	11,568,021.94	235.22	43.27	2.06	0	0	11,570,222.91
11	2,580.56	11,644,204.60	192.76	260.94	1.74	0	0	11,647,240.59
12	1,620.35	11,590,462.43	137.36	∞	0	0	∞	∞
13	2,460.53	11,628,907.01	164.3	793.14	1.48	0	0	11,632,326.46
14	2,580.56	11,547,465.29	187.32	204.44	1.64	0	0	11,550,439.25
15	3,240.70	11,633,645.73	154.22	∞	0	0	∞	∞

10.7.4 Examples of Cost Evaluations

This section illustrates how to compute each cost component for State 6 at Stage 19. The AEC of State 6 is computed in Example 10-1. In addition, The AEC of State 6 at other stages is the same as in this example. The performance penalties of Bus S1-MCC_13 at State 6, Stage 19 are computed in Example 10-2. The computational procedures for the performance penalties of the other buses are the same as those in Example 10-2. The performance penalties at State 6, Stage 19 are the sum of the performance penalties of all the load buses.

Example 10-1: This example shows the computational procedure of The AEC at State 6.

Given: The installed equipment at State 6 includes two capacitor banks: 900 $kVar$ and 1,500 $kVar$. Acquisition cost: $A_c = \$18,000$, installation cost: $I_c = \$12,000$, and the interest rate of each stage: $r = 4\%$.

Find: AEC at State 6.

Solution:

$$\begin{aligned} J_{AnnualizedEquivalentCost} &= (A_c + I_c) \left(r + \frac{r}{(1+r)^{m/p} - 1} \right) \\ &= (18,000 + 12,000) \left(0.04 + \frac{0.04}{(1+0.04)^{14/0.5} - 1} \right) \\ &= \$1,800.39 \end{aligned}$$

Example 10-2: This example shows how to compute the performance penalties of Bus S1-MCC_13 at State 6, Stage 19.

Given: The simulation result, e.g., Figure 10.4 shows the Phase A voltage of Bus S1-MCC_13 at State 6, Stage 19,

The pre-fault rated voltage, $V_{ni} = 277.128V$,

The bus load rating, $S_i = 1,100kVA$, and

The cost weights, $\beta_1 = 2.0 \text{ \$/MW}$, $\beta_2 = 30 \text{ \$/MW}$, and $\beta_3 = 1.0 \text{ \$/MW}$.

Find: The performance penalties of Bus S1-MCC_13 at State 6, Stage 19.

Solution:

According to Figure 10.4,

The actual pre-fault terminal voltage, $V_{ti} = 283.271V$,

The recovery time of bus voltage, $t_{ri} = 1.09s$, and

The oscillation magnitude of bus voltage, $V_{osci} = 6.35V$.

The voltage deviation penalty,

$$J_{voltage} = \beta_1 S_i \left(\frac{V_{ti} - V_{ni}}{0.05 V_{ni}} \right)^2 = 2.0 \times 1.1 \times \left(\frac{283.271 - 277.128}{0.05 \times 277.128} \right)^2 = \$0.4324. \quad (10-1)$$

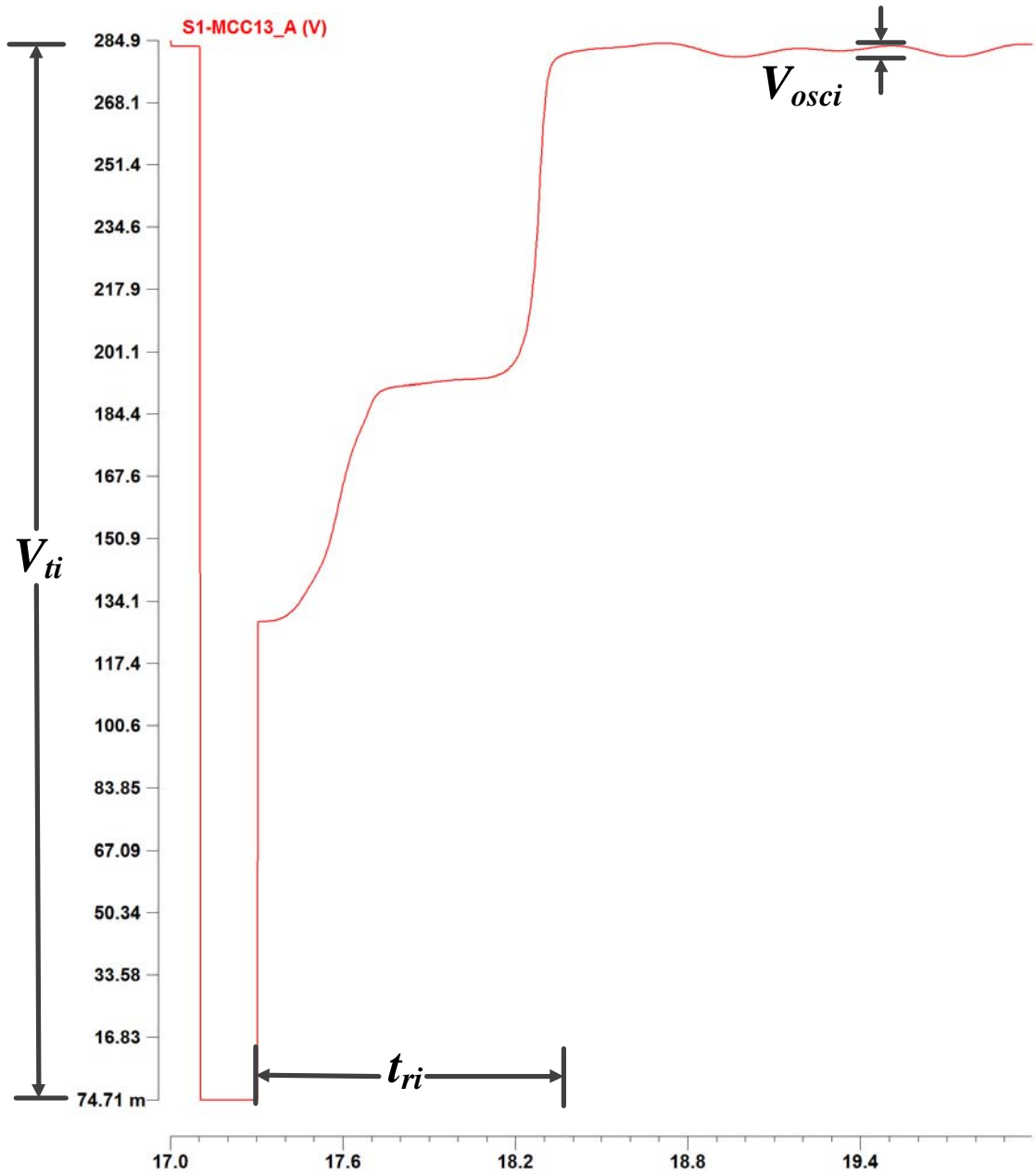


Figure 10.4 The Phase A voltage of Bus S1-MCC_13 at State 3, Stage 2

The voltage recovery time penalty,

$$J_{time} = \begin{cases} 0, & \text{if } t_{ri} < 0.5s \\ \beta_2 S_i \left(\frac{t_{ri} - 0.5}{0.5} \right)^2, & \text{if } t_{ri} \geq 0.5s \end{cases} = 30 \times 1.1 \times \left(\frac{1.09 - 0.5}{0.5} \right)^2 = \$45.95. \quad (10-2)$$

The voltage oscillation penalty,

$$\begin{aligned}
J_{oscillation} &= \begin{cases} 0, & \text{if } V_{osci} < 0.02V_{ni} \\ \beta_3 \left(\frac{V_{osci} - 0.02V_{ni}}{0.02V_{ni}} \right)^2, & \text{if } V_{osci} \geq 0.02V_{ni} \end{cases} \\
&= \begin{cases} 0, & \text{if } V_{osci} < 0.02V_{ni} \\ 1.0 \times \left(\frac{6.35 - 0.02 \times 277.128}{0.02 \times 277.128} \right)^2, & \text{if } V_{osci} \geq 0.02V_{ni} \end{cases} \quad (10-3) \\
&= \$0.0212
\end{aligned}$$

10.8 The Computational Example of Dynamic Programming

Example 10-3: This example shows how to compute the optimal trajectory cost from Stage 18 to State 6, Stage 19 using dynamic programming.

Given: Table 10-9 gives the cost data of State 6 at Stage 19. $J_{AnnualizedEquivalentCost}$ are the results of Example 10-1. $J_{VoltageDeviationPenalty}$, $J_{VoltageRecoveryTimePenalty}$, and $J_{VoltageOscillationPenalty}$ are the sum of the voltage deviation penalties, the voltage recovery time penalties, and the voltage oscillation penalties for all the load buses, respectively. Their computational procedures of Bus S1-MCC_13 are shown in Example 10-2. Both hard constraints are inactive.

Table 10-9: Cost data for State 6 at Stage 19

State #	Annualized Equivalent Cost	Operating Cost	Voltage Deviation Penalty	Voltage Recovery Time Penalty	Voltage Oscillation Penalty	Voltage Lower Bound	Voltage Recovery Time Upper Bound	State Cost
6	1,800.39	11,553,587.21	202.28	150.47	1.62	0	0	11,555,741.98

Table 10-10: Optimal trajectory costs from Stage 0 to all states at Stage 18

#	0	1	2	3
	∞	202,403,891.43	202,412,342.72	∞
#	4	5	6	7
	202,522,214.92	202,369,139.11	202,422,626.83	202,393,506.93
#	8	9	10	11
	202,490,446.66	202,403,648.74	202,436,987.82	202,415,162.85
#	12	13	14	15
	∞	202,373,416.80	202,425,689.67	202,410,100.88

Find: The optimal trajectory cost from Stage 0 to State 6, Stage 19.

Solution: The cost of State 6 at Stage 19:

$$\begin{aligned}
 C(X_{6,19}) &= J_{AnnualizedEquivalentCost}(X_{6,19}) + J_{OperatingCost}(X_{6,19}) + J_{VoltageDeviationPenalty}(X_{6,19}) + \\
 &J_{VoltageRecoveryTimePenalty}(X_{6,19}) + J_{VoltageOscillationPenalty}(X_{6,19}) + J_{VoltageLowerBound}(X_{6,19}) + \\
 &J_{VoltageRecoveryTimeUpperBound}(X_{6,19}) \\
 &= 1,800.39 + 11,553,587.21 + 202.28 + 150.47 + 1.62 \\
 &= \$11,555,741.98.
 \end{aligned} \tag{10-4}$$

The optimal trajectory cost from Stage 0 to State 6 at Stage 19:

$$\begin{aligned}
 C^*(X_{6,19}) &= \min_{\text{all state } X_{j,18} \text{ in stage 18}} \left[C^*(X_{j,18}) + T(X_{j,18} \rightarrow X_{6,19})C(X_{6,19}) \right] \\
 &= \$213,968,084.70
 \end{aligned} \tag{10-5}$$

Optimal trajectory costs from Stage 0 to other states can be computed similarly.

10.9 The Planning Details of Optimal VAR Allocation

Figure 10.5 - Figure 10.7 show the planning details. Each cell represents a state with its state #, its stage #, the state cost, and the optimal trajectory cost from Stage 0. The orange cells represent the optimal trajectory to the final stage.

0,0	0,1	0,2	0,3	0,4	0,5	0,6
0	10,991,296.09	11,017,184.14	11,046,448.45	11,075,869.72	11,106,227.94	11,136,688.72
0	10,991,296.09	22,008,480.23	33,054,928.68	44,130,798.40	55,237,026.34	66,373,715.06
	1,1	1,2	1,3	1,4	1,5	1,6
	11,009,811.49	11,035,150.26	11,060,862.80	11,082,932.19	11,109,420.59	11,136,281.86
	11,009,811.49	22,026,446.35	33,069,343.03	44,137,860.87	55,240,218.99	66,373,308.20
	2,1	2,2	2,3	2,4	2,5	2,6
	11,020,459.79	11,045,983.46	11,071,875.29	11,098,128.04	11,124,675.72	11,148,275.28
	11,020,459.79	22,037,279.55	33,080,355.52	44,153,056.72	55,255,474.12	66,385,301.62
	3,1	3,2	3,3	3,4	3,5	3,6
	11,005,628.16	11,027,491.07	11,052,535.01	11,078,450.60	11,104,838.02	11,131,621.53
	11,005,628.16	22,018,787.16	33,061,015.24	44,133,379.28	55,235,636.42	66,367,257.95
	4,1	4,2	4,3	4,4	4,5	4,6
	11,009,779.31	11,035,276.29	11,060,954.35	11,082,978.21	11,109,490.86	11,136,377.19
	11,009,779.31	22,026,572.38	33,069,434.58	44,137,906.89	55,240,289.26	66,373,403.53
	5,1	5,2	5,3	5,4	5,5	5,6
	11,025,968.78	11,051,608.41	11,077,595.74	11,103,942.95	11,130,651.76	11,157,711.87
	11,025,968.78	22,042,904.50	33,086,075.97	44,158,871.63	55,261,450.17	66,394,738.21
	6,1	6,2	6,3	6,4	6,5	6,6
	11,038,860.17	11,064,571.43	11,088,983.35	11,115,581.54	11,142,353.53	11,169,592.36
	11,038,860.17	22,055,867.52	33,097,463.59	44,170,510.22	55,273,151.93	66,406,618.70
	7,1	7,2	7,3	7,4	7,5	7,6
	11,021,591.79	11,047,045.47	11,072,959.31	11,099,234.51	11,125,864.62	11,148,877.37
	11,021,591.79	22,038,341.56	33,081,439.55	44,154,163.19	55,256,663.02	66,384,513.78
	8,1	8,2	8,3	8,4	8,5	8,6
	11,020,211.91	11,045,679.65	11,071,608.50	11,097,990.55	11,124,545.41	11,151,639.95
	11,020,211.91	22,036,975.74	33,080,088.73	44,152,919.23	55,255,343.81	66,388,666.29
	9,1	9,2	9,3	9,4	9,5	9,6
	11,035,799.73	11,064,265.47	11,088,689.72	11,115,207.94	11,142,091.68	11,169,348.95
	11,035,799.73	22,055,561.57	33,097,169.95	44,170,136.63	55,272,890.08	66,406,375.29
	10,1	10,2	10,3	10,4	10,5	10,6
	11,049,947.12	11,075,824.39	11,102,163.68	11,128,916.12	11,154,120.94	11,181,541.25
	11,049,947.12	22,067,120.49	33,110,643.91	44,183,844.80	55,284,919.34	66,418,567.59
	11,1	11,2	11,3	11,4	11,5	11,6
	11,033,899.97	11,057,828.33	11,083,906.67	11,110,343.07	11,137,146.79	11,164,323.73
	11,033,899.97	22,049,124.42	33,092,386.90	44,165,271.75	55,267,945.19	66,399,960.15
	12,1	12,2	12,3	12,4	12,5	12,6
	11,001,849.48	11,030,958.76	11,052,609.52	11,078,539.62	11,105,810.17	11,132,651.48
	11,001,849.48	22,022,254.86	33,061,089.76	44,133,468.30	55,236,608.57	66,369,260.05
	13,1	13,2	13,3	13,4	13,5	13,6
	11,021,569.60	11,047,195.78	11,073,012.40	11,099,280.55	11,125,902.72	11,148,904.94
	11,021,569.60	22,038,491.87	33,081,492.63	44,154,209.24	55,256,701.12	66,385,513.51
	14,1	14,2	14,3	14,4	14,5	14,6
	11,034,284.92	11,058,196.58	11,084,190.65	11,110,664.17	11,137,444.19	11,164,595.18
	11,034,284.92	22,049,492.67	33,092,670.89	44,165,592.85	55,268,242.59	66,401,203.74
	15,1	15,2	15,3	15,4	15,5	15,6
	11,017,138.46	11,042,707.07	11,068,448.86	11,094,738.68	11,117,135.68	11,144,092.34
	11,017,138.46	22,034,003.16	33,076,929.09	44,149,667.36	55,247,934.08	66,379,728.76

Figure 10.5 The optimal transitions from Stage 0 to Stage 6

0,7	0,8	0,9	0,10	0,11	0,12	0,13
11,168,182.83	11,200,234.93	11,232,960.41	11,266,376.57	11,300,514.19	11,335,386.98	11,370,990.76
77,541,897.90	88,742,132.83	99,975,093.23	111,241,469.80	122,541,984.00	133,877,370.98	145,248,361.74
1,7	1,8	1,9	1,10	1,11	1,12	1,13
11,163,517.87	11,191,131.70	11,219,125.39	11,251,186.61	11,277,069.74	11,309,442.70	11,344,967.16
77,536,826.07	88,727,957.77	99,947,083.16	111,198,269.77	122,475,339.51	133,784,782.20	145,129,749.36
2,7	2,8	2,9	2,10	2,11	2,12	2,13
11,175,335.20	11,203,808.85	11,231,044.20	11,259,583.13	11,288,507.15	11,317,817.43	11,347,516.00
77,549,050.26	88,745,706.75	99,973,177.03	111,232,760.16	122,521,267.31	133,839,084.74	145,186,600.74
3,7	3,8	3,9	3,10	3,11	3,12	3,13
11,158,778.88	11,187,210.28	11,217,260.76	11,250,621.13	11,284,047.14	11,318,658.25	11,353,607.78
77,526,036.83	88,713,247.11	99,930,507.87	111,181,129.00	122,465,176.14	133,783,834.40	145,137,442.18
4,7	4,8	4,9	4,10	4,11	4,12	4,13
11,163,640.01	11,192,166.69	11,221,514.43	11,254,983.63	11,288,534.80	11,323,052.13	11,358,088.49
77,537,043.54	88,729,210.23	99,950,724.66	111,205,708.29	122,494,243.08	133,817,295.21	145,175,383.70
5,7	5,8	5,9	5,10	5,11	5,12	5,13
11,181,804.23	11,209,154.29	11,237,421.09	11,266,072.80	11,295,111.30	11,324,538.77	11,354,355.87
77,555,112.43	88,745,980.36	99,965,378.86	111,213,155.96	122,493,381.07	133,799,878.28	145,139,138.07
6,7	6,8	6,9	6,10	6,11	6,12	6,13
11,197,206.68	11,225,278.17	11,249,830.65	11,278,650.44	11,307,859.01	11,337,456.23	11,367,445.15
77,570,610.21	88,762,321.71	99,979,040.88	111,229,375.10	122,513,567.30	133,831,699.31	145,184,740.36
7,7	7,8	7,9	7,10	7,11	7,12	7,13
11,176,299.19	11,204,101.45	11,232,285.64	11,260,854.14	11,289,809.48	11,323,379.66	11,348,884.81
77,543,557.14	88,730,138.28	99,945,532.75	111,191,362.01	122,470,938.48	133,788,555.80	145,132,719.20
8,7	8,8	8,9	8,10	8,11	8,12	8,13
11,175,020.38	11,203,408.75	11,231,045.98	11,259,635.18	11,288,612.12	11,318,708.26	11,349,907.08
77,548,735.44	88,745,306.65	99,973,178.81	111,232,813.99	122,521,426.11	133,840,134.37	145,190,041.44
9,7	9,8	9,9	9,10	9,11	9,12	9,13
11,196,981.75	11,225,068.64	11,249,647.05	11,278,487.93	11,307,717.65	11,339,685.53	11,367,350.02
77,570,289.96	88,761,894.71	99,977,604.82	111,225,571.09	122,505,987.42	133,815,025.04	145,152,132.22
10,7	10,8	10,9	10,10	10,11	10,12	10,13
11,209,323.41	11,237,490.88	11,266,038.05	11,295,214.79	11,321,031.85	11,350,606.08	11,381,126.24
77,583,038.47	88,779,388.78	100,008,170.87	111,268,391.81	122,553,792.01	133,871,873.39	145,220,210.98
11,7	11,8	11,9	11,10	11,11	11,12	11,13
11,187,991.49	11,215,977.03	11,244,346.54	11,273,102.73	11,302,247.44	11,331,781.85	11,361,707.95
77,555,249.44	88,742,013.86	99,957,593.64	111,203,610.60	122,483,376.44	133,796,958.00	145,145,542.35
12,7	12,8	12,9	12,10	12,11	12,12	12,13
11,161,630.53	11,193,746.49	11,226,107.09	11,259,550.88	11,293,234.28	11,327,914.46	11,363,252.87
77,530,890.57	88,724,637.06	99,950,744.15	111,210,295.03	122,503,529.31	133,831,443.77	145,194,696.64
13,7	13,8	13,9	13,10	13,11	13,12	13,13
11,176,597.42	11,204,110.58	11,232,285.12	11,260,843.88	11,289,788.18	11,319,120.15	11,348,841.33
77,545,857.46	88,735,001.15	99,956,922.19	111,207,927.04	122,488,057.94	133,794,459.65	145,133,623.54
14,7	14,8	14,9	14,10	14,11	14,12	14,13
11,188,231.30	11,216,188.26	11,244,528.35	11,273,253.68	11,302,366.21	11,331,867.10	11,361,757.90
77,557,491.35	88,747,078.83	99,969,165.41	111,223,997.83	122,512,661.24	133,835,396.41	145,193,201.67
15,7	15,8	15,9	15,10	15,11	15,12	15,13
11,171,906.28	11,199,136.73	11,227,230.05	11,255,706.50	11,288,472.86	11,313,817.05	11,346,747.42
77,539,164.23	88,725,173.56	99,940,477.16	111,186,214.37	122,469,601.86	133,778,993.20	145,125,740.62

Figure 10.6 The optimal transitions from Stage 7 to Stage 13

0,14	0,15	0,16	0,17	0,18	0,19	0,20
11,407,353.66	11,444,634.07	11,482,882.11	11,589,069.33	∞	∞	∞
156,655,715.39	168,100,349.46	179,583,231.58	191,172,300.91	∞	∞	∞
1,14	1,15	1,16	1,17	1,18	1,19	1,20
11,380,240.00	11,416,699.32	11,453,922.76	11,492,065.51	11,531,214.49	∞	∞
156,509,989.36	167,926,688.68	179,380,611.43	190,872,676.94	202,403,891.43	∞	∞
2,14	2,15	2,16	2,17	2,18	2,19	2,20
11,377,604.18	11,408,085.43	11,447,352.13	11,479,688.87	11,513,011.37	11,630,500.74	∞
156,564,204.92	167,972,290.35	179,419,642.48	190,899,331.35	202,412,342.72	214,042,843.46	∞
3,14	3,15	3,16	3,17	3,18	3,19	3,20
11,389,532.22	11,426,183.80	11,463,650.40	11,502,077.50	∞	∞	∞
156,526,974.40	167,953,158.19	179,416,808.59	190,918,886.09	∞	∞	∞
4,14	4,15	4,16	4,17	4,18	4,19	4,20
11,394,098.16	11,430,842.52	11,468,401.40	11,506,838.40	11,546,650.72	∞	∞
156,569,481.87	168,000,324.39	179,468,725.79	190,975,564.20	202,522,214.92	∞	∞
5,14	5,15	5,16	5,17	5,18	5,19	5,20
11,390,159.07	11,415,166.43	11,446,168.70	11,479,973.63	11,517,841.00	11,638,638.23	∞
156,519,908.42	167,925,155.79	179,371,324.49	190,851,298.11	202,369,139.11	214,007,777.34	∞
6,14	6,15	6,16	6,17	6,18	6,19	6,20
11,402,266.10	11,434,028.96	11,459,767.38	11,491,330.17	11,523,295.48	11,555,741.98	11,688,756.36
156,577,649.80	167,998,233.87	179,432,057.73	190,910,972.65	202,422,626.83	213,968,084.70	225,656,841.06
7,14	7,15	7,16	7,17	7,18	7,19	7,20
11,379,606.76	11,413,778.86	11,450,939.22	11,488,905.05	11,527,557.84	11,632,715.93	∞
156,512,325.97	167,926,104.83	179,377,044.05	190,865,949.10	202,393,506.93	214,026,222.86	∞
8,14	8,15	8,16	8,17	8,18	8,19	8,20
11,385,175.15	11,422,139.29	11,459,319.28	11,497,363.96	11,536,407.54	∞	∞
156,575,216.59	167,997,355.88	179,456,675.16	190,954,039.12	202,490,446.66	∞	∞
9,14	9,15	9,16	9,17	9,18	9,19	9,20
11,397,755.84	11,428,554.99	11,459,749.51	11,491,342.29	11,531,695.02	11,565,248.03	11,689,010.75
156,527,505.20	167,938,544.35	179,386,438.18	190,871,953.72	202,403,648.74	213,968,896.77	225,657,907.52
10,14	10,15	10,16	10,17	10,18	10,19	10,20
11,411,375.12	11,442,351.25	11,473,722.87	11,505,490.61	11,537,656.47	11,570,222.91	11,704,552.85
156,597,975.86	168,006,556.17	179,446,013.22	190,925,133.09	202,436,987.82	213,982,565.63	225,687,118.47
11,14	11,15	11,16	11,17	11,18	11,19	11,20
11,396,845.34	11,422,739.17	11,453,852.12	11,485,370.87	11,526,226.31	11,647,240.59	∞
156,534,287.52	167,949,713.56	179,403,565.68	190,888,936.54	202,415,162.85	214,062,403.45	∞
12,14	12,15	12,16	12,17	12,18	12,19	12,20
11,399,364.61	11,436,245.84	11,473,948.57	11,512,709.45	∞	∞	∞
156,594,061.25	168,030,307.09	179,504,255.66	191,016,965.11	∞	∞	∞
13,14	13,15	13,16	13,17	13,18	13,19	13,20
11,378,954.27	11,409,911.80	11,446,587.55	11,485,025.19	11,523,188.64	11,632,326.46	∞
156,508,703.63	167,918,615.43	179,365,202.97	190,850,228.16	202,373,416.80	214,005,743.26	∞
14,14	14,15	14,16	14,17	14,18	14,19	14,20
11,392,040.40	11,422,714.41	11,453,781.99	11,485,249.52	11,526,358.33	11,550,439.25	∞
156,578,641.13	167,986,919.33	179,426,072.34	190,904,892.00	202,425,689.67	213,962,781.97	∞
15,14	15,15	15,16	15,17	15,18	15,19	15,20
11,382,390.40	11,418,954.68	11,456,070.44	11,493,965.27	11,532,979.49	∞	∞
156,508,131.01	167,927,085.69	179,383,156.13	190,877,121.39	202,410,100.88	∞	∞

Figure 10.7 The optimal transitions from Stage 14 to Stage 20

The optimal trajectory shows that the best choice is installing a 1,500 *Var* capacitor bank at Location 2 at Stage 9 and installing a 900 *Var* capacitor bank at Location 1 at Stage 19. If the final stage is Stage 19, the best choice will be installing a 600 *Var* SVC at Stage 19 instead of a 900 *Var* capacitor bank. Table 10-11 shows the back tracing information providing the optimal trajectory of any states. In this table, each row represents a state and each column represents a stage. The number in Cell i, j (i , = the state index, j = the stage index) gives the state index in Stage $j - 1$ on the optimal trajectory of State i , Stage j . For instance, State 1, Stage 16 is on the optimal trajectory of State 9, Stage 17. Orange cells represent the optimal trajectory from Stage 0 to Stage 20.

Table 10-11: Back tracing information of dynamic programming

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	-1	-1
1		0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	-1	-1
2		0	0	0	0	0	0	0	0	0	2	2	2	2	2	2	2	2	2	2	-1
3		0	0	0	0	0	3	3	3	3	3	3	3	3	3	3	3	3	-1	-1	-1
4		0	0	0	0	0	0	4	4	4	4	4	4	4	4	4	4	4	4	-1	-1
5		0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	5	5	5	5	-1
6		0	0	0	0	0	0	4	4	4	4	4	4	4	4	2	2	2	2	2	6
7		0	0	0	0	0	3	3	3	3	3	3	3	3	7	7	7	7	7	7	-1
8		0	0	0	0	0	0	0	0	0	8	8	8	8	8	8	8	8	8	-1	-1
9		0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	9	9	9
10		0	0	0	0	0	0	0	0	0	2	2	2	2	2	2	2	2	2	2	10
11		0	0	0	0	0	3	3	3	3	3	3	3	3	3	3	11	11	11	11	-1
12		0	0	0	0	0	12	12	12	12	12	12	12	12	12	12	12	12	-1	-1	-1
13		0	0	0	0	0	12	12	12	12	1	1	1	1	1	13	13	13	13	13	-1
14		0	0	0	0	0	12	12	12	12	12	12	12	12	2	2	2	2	2	2	-1
15		0	0	0	0	0	3	3	3	3	3	3	3	15	15	15	15	15	15	-1	-1

10.10 Summary

This chapter gave an example that illustrates how the dynamic programming algorithm can be used to perform VAR planning on a test distribution network. The network is simulated on the software WinIGS – Q. TOPF is used to compute the

operating costs for all states in all stages. The planning result shows that the operating costs computed via TOPF are a significant part in the state cost.

CHAPTER 11

CONCLUSIONS AND FUTURE RESEARCH DIRECTION

11.1 Conclusions

The work performed in this thesis is to develop a robust and high-efficient OPF algorithm via the quadratic model on both single-phase and three-phase power systems. The proposed algorithm can solve power systems with various types of devices. A special type of devices proposed in this work is the mismatch current source at each bus, which can help the algorithm starting from arbitrary states which can be feasible or infeasible. Therefore, this algorithm can even provide a solution for infeasible systems (without a valid power flow solution). The result of an infeasible system will have nonzero mismatches at the final step and these mismatches can be converted to remedial actions such as load shedding.

The models of single-phase and three-phase power systems in this thesis are both quadratized for excellent convergence properties. Quadratzation is achieved via using Cartesian co-ordinates and introducing additional internal state variables. A device in the power system usually has control variables and state variables classified according to device types and modes. For example, synchronous generators in slack mode, PQ mode, and PV mode have different variable classifications. The proposed OPF algorithm solves for the control variables in the optimization step and for the state variables in the power flow step.

This work can identify active operating constraints and add only them to the model via adding the violated constraints of the previous iteration. For example, the three-phase RTS-96 system has 1,631 operating constraints, while the number of final active constraints added is only 275, which is 16.86% of the total amount.

Sparsity technologies can highly reduce the storage and runtime in linearization and solving the power flow. If the algorithm does not use sparsity technologies, the size of the Jacobian matrix is $O(N^2)$ for a power system with N state variables. The computation of the inverse Jacobian matrix requires $O(N^3)$ runtime when using a common inverse matrix algorithm. For example, the three-phase RTS-96 system has 2,360 state variables that require over 10^{10} computations. Sparsity technologies can solve this problem via manipulating only nonzero cells since most of the cells in the Jacobian matrix are filled with 0.

The developed OPF software has been tested on eleven systems sized from three buses to three hundred buses and the results are compared with seven well-known OPF software packages. The TOPF software is also tested on four three-phase systems.

Finally, an important contribution of this work is applying the proposed TOPF algorithm on a planning problem: optimal VAR allocation mitigating FIDVR phenomena. This problem is solved via dynamic programming and TOPF is used to compute the operating costs in each planning stage. This part demonstrates the use of OPF in power system planning.

11.2 Future Work

This thesis work has proposed an OPF algorithm and tested on various benchmark systems that contain up to 300 buses. However, OPF is a complicated concave optimization problem and should be applied to large-scale power systems. There are several research topics related to this thesis, which can be investigated in the future.

A first extension is to apply the algorithm on large-scale systems since power systems nowadays include over tens of thousands of buses and control variables. This task may need further optimization of the code. First, the present code linearizes the operating constraints separately. Each linearization process requires solving the linear equation system $Ax = b$ once. Since A does not change in one iteration, an advanced

solving method is expected. Second, sequential methods should to solve the power flow when updating state variables. This step occurs in every Layer-3 loop. A faster power flow code will reduce the runtime tremendously. Third, the present code parallelizes the linearization step. Parallelism on other time consuming steps such as LP and power flow also requires additional work.

A second extension will be a smarter method to manage the modeled operating constraints. The present algorithm keeps adding operating constraints if they are violated in the previous iteration. Since working points move during iteration times, some constraints may not reach their boundaries afterwards. Future work can add one or more steps to remove some modeled operating constraints.

An important extension is the further improvement of the convergence speed. The convergence speed depends on the moving direction of the working point and the step lengths of the control variables. In the present algorithm version, the moving direction is determined by LP or QP, and the step lengths are controlled by the linearization limits, which is an important topic in SLP or SQP. Selecting larger limits means a faster convergence but a larger oscillation, while selecting smaller limits means a slower convergence speed. The given algorithm computes the linearization limits based on the linearization error of the power balance equations or the current conservation equations. The inequity constraints are not included in the computation of the limits due to less usefulness and a larger computation burden. Some advanced step-length strategies need to be developed since a high convergence speed is very important for large-scale systems.

Another important extension of the presented work lies in the further development of the TOPF software. The device models in the present TOPF code contain the control variables indirectly. Therefore, the partial derivatives of the power flow equations with respect to the control variables are computed via a sensitivity analysis, which is much slower than a direct substitution. Since OPF and TOPF both require high computational

efficiency, their code should be greatly optimized before releasing it for practical or commercial use.

An evident further extension is a better planning algorithm in optimal VAR allocation, which is also a NP-complete problem. The proposed algorithm reduces the search space via preselecting candidate locations using sensitivity analysis. However, this method is not accurate and may not find better selections. An obvious improvement is repeating the given method on several location combinations and then selecting the one with the minimum cost.

Finally, several investigations of the proposed OPF algorithm and the quadratized system model are of interest. This dissertation work demonstrated and compared the sequential OPF algorithm with several other methods. One of the observations is that NLP methods using the Kuhn-Tucker conditions also promise a good efficiency. Therefore, future work can develop a NLP method using the quadratized model. In addition, people can also extend this algorithm to SCOPF including the contingency constraints.

APPENDIX A

QUADRATIC SINGLE-PHASE TRANSFORMER MODEL

This section describes the model of the single-phase two-winding transformer. Three of these models can be connected into the four subcases of Y-Y, Y- Δ , Δ -Y, and Δ - Δ connected transformers.

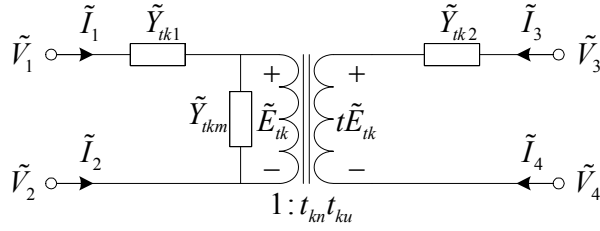


Figure A.1 The single-phase transformer model

Figure A.1 illustrates the physical model of the single-phase variable-tap transformer. The turn ratio t_k consists of two parts: one is the nominal transformation ratio t_{kn} and the other is the per-unit tap selection t_{ku} . The overall turn ratio $t_k = t_{kn} t_{ku}$. The admittance \tilde{Y}_{tk1} and \tilde{Y}_{tk2} of the transformer are expressed as follows:

$$\tilde{Y}_{tk1} = 2\tilde{Y}_{tk} t_{kn}^2 = \frac{1}{r_{k1} + j\omega L_{k1}} \quad \text{and} \quad (A.1)$$

$$\tilde{Y}_{tk2} = \frac{2\tilde{Y}_{tk}}{1 + \text{abs}(1 - t_{ku})} = \frac{1}{r_2 + j\omega L_2}, \quad (A.2)$$

where $\tilde{Y}_{tk} = g_{tk} + jb_{tk}$ is the nominal admittance of the transformer referred to the secondary side. $\tilde{Y}_{tkm} = g_{tkm} + jb_{tkm}$ is the magnetizing admittance referred to the primary side. This model assumes that the leakage impedance is proportional to the number of turns.

According to Kirchhoff's laws, the equations of a single-phase transformer are

$$\tilde{I}_1 = \tilde{Y}_{tk1} (\tilde{V}_1 - \tilde{V}_2 - \tilde{E}_{tk}), \quad (A.3)$$

$$\tilde{I}_3 = \tilde{Y}_{tk2} (\tilde{V}_3 - \tilde{V}_4 - t_{kn} t_{ku} \tilde{E}_{tk}), \quad \text{and} \quad (A.4)$$

$$0 = -\tilde{Y}_{tk1} (\tilde{V}_1 - \tilde{V}_2 - \tilde{E}_{tk}) - t_{kn} t_{ku} \tilde{Y}_{tk2} (\tilde{V}_3 - \tilde{V}_4 - t_{kn} t_{ku} \tilde{E}_{tk}) + \tilde{Y}_{tkm} \tilde{E}_{tk}. \quad (\text{A.5})$$

Two state variables $u_{tk1} = \sqrt{(1-t_{ku})^2}$ and $u_{tk2} = \frac{1}{1+u_{tk1}}$ are introduced to eliminate

the absolute value function in the denominator in \tilde{Y}_{tk2} . Other two state variables

$u_{tk3} = u_{tk2} t_{ku}$ and $u_{tk4} = u_{tk3} t_{ku}$ are introduced to reduce the order of the equation system.

By substituting them into (A.1) to (A.5) and separating the equations into real and imaginary parts, the following quadratized equations of a single-phase transformer are formed:

$$I_{1r} = 2t_{kn}^2 g_{tk} (V_{1r} - V_{2r} - E_{tkr}) - 2t_{kn}^2 b_{tk} (V_{1i} - V_{2i} - E_{tki}), \quad (\text{A.6})$$

$$I_{1i} = 2t_{kn}^2 g_{tk} (V_{1i} - V_{2i} - E_{tki}) + 2t_{kn}^2 b_{tk} (V_{1r} - V_{2r} - E_{tkr}), \quad (\text{A.7})$$

$$I_{3r} = 2u_{tk2} g_{tk} (V_{3r} - V_{4r}) - 2u_{tk2} b_{tk} (V_{3i} - V_{4i}) - 2g_{tk} t_{kn} u_{tk3} E_{tkr} + 2b_{tk} t_{kn} u_{tk3} E_{tki}, \quad (\text{A.8})$$

$$I_{3i} = 2u_{tk2} g_{tk} (V_{3i} - V_{4i}) + 2u_{tk2} b_{tk} (V_{3r} - V_{4r}) - 2g_{tk} t_{kn} u_{tk3} E_{tki} - 2b_{tk} t_{kn} u_{tk3} E_{tkr}, \quad (\text{A.9})$$

$$\begin{aligned} 0 = & 2t_{kn}^2 g_{tk} (V_{1i} - V_{2i}) + 2t_{kn}^2 b_{tk} (V_{1r} - V_{2r}) \\ & + 2t_{kn} g_{tk} u_{tk3} (V_{3i} - V_{4i}) + 2t_{kn} b_{tk} u_{tk3} (V_{3r} - V_{4r}) \\ & - (2t_{kn}^2 g_{tk} + 2t_{kn}^2 g_{tk} u_{tk4} + g_{tmk}) E_{tki} - (2t_{kn}^2 b_{tk} + 2t_{kn}^2 b_{tk} u_{tk4} + b_{tmk}) E_{tkr} \end{aligned}, \quad (\text{A.10})$$

$$\begin{aligned} 0 = & 2t_{kn}^2 g_{tk} (V_{1r} - V_{2r}) - 2t_{kn}^2 b_{tk} (V_{1i} - V_{2i}) \\ & + 2t_{kn} g_{tk} u_{tk3} (V_{3r} - V_{4r}) - 2t_{kn} b_{tk} u_{tk3} (V_{3i} - V_{4i}) \\ & - (2t_{kn}^2 g_{tk} + 2t_{kn}^2 g_{tk} u_{tk4} + g_{tmk}) E_{tkr} + (2t_{kn}^2 b_{tk} + 2t_{kn}^2 b_{tk} u_{tk4} + b_{tmk}) E_{tki} \end{aligned}, \quad (\text{A.11})$$

$$0 = 2t_{ku} - t_{ku}^2 + u_{tk1}^2 - 1, \quad (\text{A.12})$$

$$0 = u_{tk2} + u_{tk1} u_{tk2} - 1, \quad (\text{A.13})$$

$$0 = u_{tk2} t_{ku} - u_{tk3}, \text{ and} \quad (\text{A.14})$$

$$0 = u_{tk3} t_{ku} - u_{tk4}, \quad (\text{A.15})$$

This model includes six state variables $[E_{tk1r}, E_{tk1i}, u_{tk1}, u_{tk2}, u_{tk3}, u_{tk4}]$ and one control variable t_{ku} .

APPENDIX B

LINEARIZATION METHODS

B.1 Overview

The original OPF problem is a nonlinear optimization problem. The state variables are eliminated and the problem is re-cast in terms of control variables only. This is achieved by linearization where all functions/quantities are expressed as a linear combination of control variables. Therefore, the nonlinear optimization problem is reformulated to a linear optimization problem and can be solved by well-designed LP algorithms.

Theoretically, the coefficient of a control variable in the linearized function equals the total derivative of the original function with respect to the control variable. The state variables are eliminated in the linearization step since they can be solved according to the power flow equations. However, if a nonlinear equation is a conditional identity according to the power flow equations, such as power balance equations and current conservation equations, their total derivatives with respect to any control variable are zero. The explanation is as follows: $f(\mathbf{x}, \mathbf{u}, v) = 0$ is assumed to be the real power balance equation. The total derivative of $f(\mathbf{x}, \mathbf{u}, v)$ with respect to Δu_i is $\frac{f(\mathbf{x}^1, \mathbf{u}^1, 0) - f(\mathbf{x}^o, \mathbf{u}^o, 0)}{\Delta u_i}$, where $(\mathbf{x}^1, \mathbf{u}^1, 0)$ is the state after changing u_i to $u_i + \Delta u_i$ from state $(\mathbf{x}^o, \mathbf{u}^o, 0)$. Obviously, $f(\mathbf{x}^o, \mathbf{u}^o, 0) = 0$ and $f(\mathbf{x}^1, \mathbf{u}^1, 0) = 0$ according to the power flow equations. $\frac{df(\mathbf{x}, \mathbf{u}, v)}{d\mathbf{u}} = \mathbf{0}$, because $f(\mathbf{x}^1, \mathbf{u}^1, 0) - f(\mathbf{x}^o, \mathbf{u}^o, 0) = 0$. Therefore, the total derivatives of the real power balance equation cannot reflect the rates of change of the control variables. In addition, although the partial derivatives $\frac{\partial f(\mathbf{x}, \mathbf{u}, v)}{\partial \mathbf{u}} \neq \mathbf{0}$, they cannot be used as the coefficients of the control variables since they do not have any physical meanings. To overcome these

problems, the linearization step uses the reduced power flow equations, referred to as $\mathbf{g}_{reduced}(\mathbf{x}, \mathbf{u}, v) = 0$. The reduced power flow equations consist of the power flow equations excluding all the slack bus equations in a symmetric and balanced power system or the neutral-phase equations at the slack bus in a three-phase power system.

Since no conditional identity exists, $\left. \frac{\partial f(\mathbf{x}, \mathbf{u}, v)}{\partial \mathbf{u}} \right|_{\mathbf{g}_{reduced}(\mathbf{x}, \mathbf{u}, v)=0} \neq \mathbf{0}$, which is suitable for the

use as linearized coefficients of the control variables. Linearization using the reduced power flow equations eliminates all the state variables not associated with the slack bus. The state variables of the slack bus can also be eliminated since they are assumed to be constant in the optimization step. Two methods to obtain the coefficients of the linearized functions are presented in the following two subsections.

B.2 The Definition of the Derivative

The derivative of $f(\mathbf{x}, \mathbf{u}, v)$ with respect to u_i is computed as follows,

1. Compute the value of $f(\mathbf{x}, \mathbf{u}, v)$ at the present operating point, referred to as $f(\mathbf{x}^o, \mathbf{u}^o, 0)$.
2. Change u_i infinitesimally, referred to as $u_{i1}=u_{i0}+\Delta u_i$. The value of the function $f(\mathbf{x}, \mathbf{u}, v)$

in this step is $f(\mathbf{x}^o, \mathbf{u}^1, 0)$. The partial derivative of $f(\mathbf{x}, \mathbf{u}, v)$ with respect to u_i ($\frac{\partial f(\mathbf{x}, \mathbf{u}, v)}{\partial u_i}$)

is $\frac{f(\mathbf{x}^o, \mathbf{u}^1, 0) - f(\mathbf{x}^o, \mathbf{u}^o, 0)}{\Delta u_i}$.

3. Resolve the reduced power flow equations and obtain the updated values of the state variables (\mathbf{x}^1). The function value changes to $f(\mathbf{x}^1, \mathbf{u}^1, 0)$.

$$4. \left. \frac{\partial f(\mathbf{x}, \mathbf{u}, v)}{\partial u_i} \right|_{\mathbf{g}_{reduced}(\mathbf{x}, \mathbf{u}, v)=0} = \frac{f(\mathbf{x}^1, \mathbf{u}^1, 0) - f(\mathbf{x}^o, \mathbf{u}^o, 0)}{\Delta u_i}. \quad (\text{B.1})$$

B.3 The Co-state Method

For a general operating constraint $f(\mathbf{x}, \mathbf{u}, v) \leq 0$, its linearized constraint is

$$f(\mathbf{x}^o, \mathbf{u}^o, 0) + \left. \frac{\partial f(\mathbf{x}, \mathbf{u}, v)}{\partial \mathbf{u}} \right|_{\mathbf{g}_{\text{reduced}}(\mathbf{x}, \mathbf{u}, v)=0} \cdot \Delta \mathbf{u} \leq 0. \quad (\text{B.2})$$

The partial derivative with respect to \mathbf{u} according to the reduced power flow equations is

$$\left. \frac{\partial f(\mathbf{x}, \mathbf{u}, v)}{\partial \mathbf{u}} \right|_{\mathbf{g}_{\text{reduced}}(\mathbf{x}, \mathbf{u}, v)=0} = \frac{\partial f(\mathbf{x}^o, \mathbf{u}^o, 0)}{\partial \mathbf{u}} - \frac{\partial f(\mathbf{x}^o, \mathbf{u}^o, 0)}{\partial \mathbf{x}} \left(\frac{\partial \mathbf{g}_{\text{reduced}}(\mathbf{x}^o, \mathbf{u}^o, 0)}{\partial \mathbf{x}} \right)^{-1} \frac{\partial \mathbf{g}_{\text{reduced}}(\mathbf{x}^o, \mathbf{u}^o, 0)}{\partial \mathbf{u}}, \quad (\text{B.3})$$

where

$(\mathbf{x}^o, \mathbf{u}^o, 0)$ is the current working point and

$\frac{\partial \mathbf{g}_{\text{reduced}}(\mathbf{x}^o, \mathbf{u}^o, 0)}{\partial \mathbf{x}}$ is the Jacobian matrix of the reduced power flow equations at the working point $(\mathbf{x}^o, \mathbf{u}^o, 0)$.

The co-state method requires the computation of the inverse Jacobian matrix and the multiplication of matrices. These computations need sparsity techniques since the algorithm is intended for large-scale systems.

B.4 Linearization Update Methods

The operating constraints in the OPF problem are linearized as well, but the linearized constraints cannot guarantee that the original nonlinear constraints are satisfied due to the linearization error. If a nonlinear constraint is violated, the algorithm will change its corresponding linearized constraint via the linearization update method, retrieve the previous working point, and then redo the linear programming (LP). Therefore, the nonlinear constraint will be satisfied in the steps thereafter.

The linearized constraint of $h(\mathbf{x}, \mathbf{u}) \leq h^{\max}$ is $\frac{dh(\mathbf{x}^o, \mathbf{u}^o)}{d\mathbf{u}} \cdot \Delta \mathbf{u} \leq \Delta h^{\max}$ and the linearized constraint of $h^{\min} \leq h(\mathbf{x}, \mathbf{u})$ is $\Delta h^{\min} \leq \frac{dh(\mathbf{x}^o, \mathbf{u}^o)}{d\mathbf{u}} \cdot \Delta \mathbf{u}$. This analysis focuses on the upper bound constraints only, and the formulas of the lower bound constraints can be obtained similarly.

First, we analyze a simple case: the LP result reaches the boundary of a constraint. Since the working point does not change if any violation occurs, the linearized constraints without violation remain unchanged. If any constraint is violated, a certain value should be subtracted from the corresponding b of that constraint. The overshoot ($h(\mathbf{x}, \mathbf{u}) - h^{\max}$) is subtracted from b . Therefore, the new value of b is

$$\begin{aligned} \Delta h^{\max} & \quad \text{if } h(\mathbf{x}, \mathbf{u}) \leq h^{\max} \text{ is not violated or} \\ \Delta h^{\max} - [h(\mathbf{x}, \mathbf{u}) - h^{\max}] & \quad \text{if } h(\mathbf{x}, \mathbf{u}) \leq h^{\max} \text{ is violated} \end{aligned} \quad (\text{B.4})$$

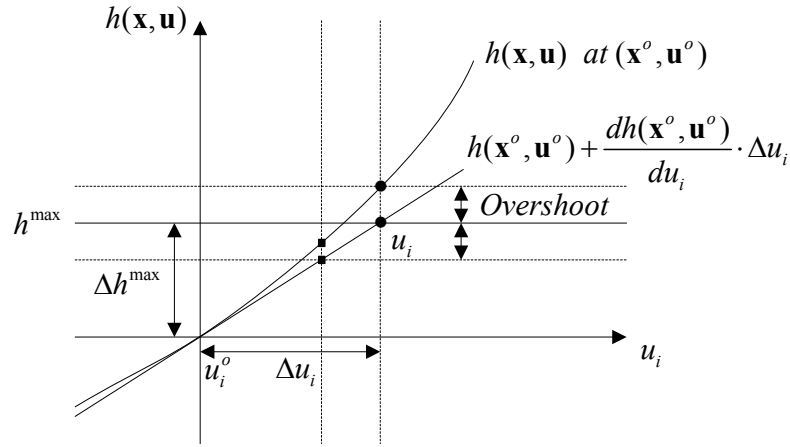


Figure B.1 The result reaches the boundary of a constraint

Figure B.1 shows the violation caused by the linearization error of the variable $u_i \in \mathbf{u}$. If the overshoot is subtracted from b , the solution moves to the left dotted vertical line. Figure B.1 shows that the constraint is satisfied. The dots represent the solution before the update and the squares represent the updated solution. However, the LP solution may not reach to the boundary of the linearized constraint, but the solution may already violate the corresponding nonlinear constraint. In this situation, Formula (B.4) cannot guarantee that the updated solution satisfies the nonlinear constraint. As shown in Figure B.1, the overshoot is subtracted from the right side of the linearized constraint. The linearized constraint changes to

$$\left. h(\mathbf{x}^o, \mathbf{u}^o) + \Delta \mathbf{u} \cdot \frac{\partial h(\mathbf{x}^o, \mathbf{u}^o)}{\partial \mathbf{u}} \right|_{\mathbf{g}_{reduced}(\mathbf{x}, \mathbf{u}, \mathbf{v})=0} \leq h^{\max} - [h(\mathbf{x}, \mathbf{u}) - h^{\max}]. \quad (\text{B.5})$$

The updated right side of the constraint is shown as the lower horizontal dashed line in Figure B.1. In this situation, $u_i^o + \Delta u_i$ or its adjacent points can be also accepted as the LP solution, but the nonlinear operating constraint $h(\mathbf{x}, \mathbf{u}) \leq h^{\max}$ will be violated again. According to the algorithm, this process continues until the right side of linearized constraint becomes low enough to ensure that the new working point satisfies the nonlinear operating constraint $h(\mathbf{x}, \mathbf{u}) \leq h^{\max}$. In some extreme cases, many updates are required to meet the nonlinear constraint.

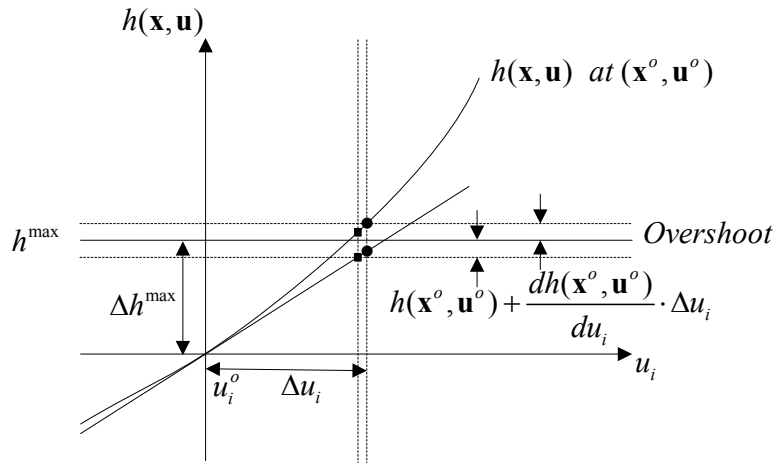


Figure B.2 The result does not reach the boundary of the constraint

To solve the problem above, the power flow update of the linearized constraint should also be subtracted from the right side of the linearized constraint. Figure B.2 shows this method. The new value of b is

$$\begin{aligned}
 & \Delta h^{\max} \quad \text{if } h(\mathbf{x}, \mathbf{u}) \leq h^{\max} \text{ is not violated or} \\
 & \Delta h^{\max} - [h(\mathbf{x}, \mathbf{u}) - h^{\max}] \\
 & - \left\{ \Delta h^{\max} - \sum_{u_i \in \mathbf{u}} \left[\left(\frac{\partial h(\mathbf{x}^o, \mathbf{u}^o)}{\partial u_i} \right) \Delta u_i \right] - \left(\frac{\partial h(\mathbf{x}^o, \mathbf{u}^o)}{\partial v} \right) v \right\} \quad \text{if } h(\mathbf{x}, \mathbf{u}) \leq h^{\max} \text{ is violated.} \quad (B.6)
 \end{aligned}$$

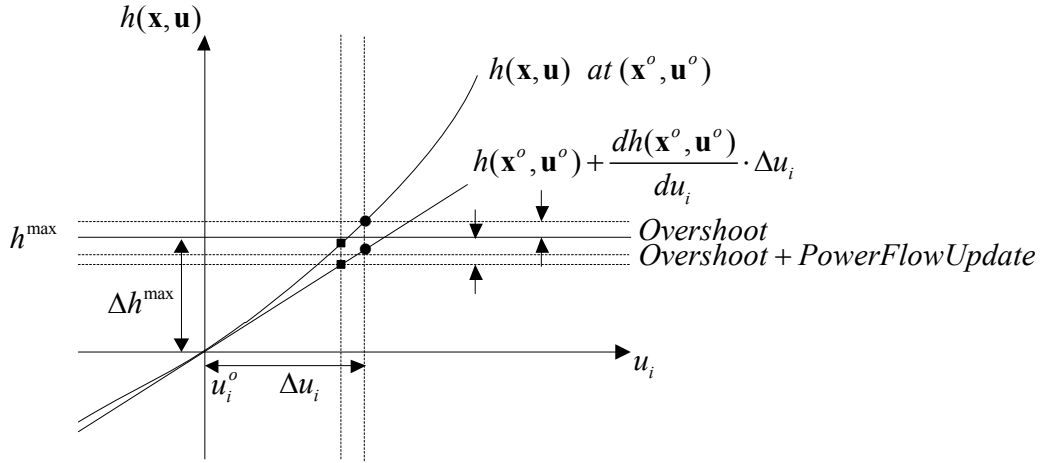


Figure B.3 The final version of the linearization update method

The updated linearized formula of the lower bound constraint can be obtained similarly. The new value of b is

$$\begin{aligned}
 & \Delta h^{\min} \quad \text{if } h^{\min} \leq h(\mathbf{x}, \mathbf{u}) \text{ is not violated or} \\
 & \Delta h^{\min} + [h^{\min} - h(\mathbf{x}, \mathbf{u})] \\
 & + \left\{ \Delta h^{\min} - \sum_{u_i \in \mathbf{u}} \left[\left(\frac{\partial h(\mathbf{x}^o, \mathbf{u}^o)}{\partial u_i} \right) \Delta u_i \right] - \left(\frac{\partial h(\mathbf{x}^o, \mathbf{u}^o)}{\partial v} \right) v \right\} \quad (B.7) \\
 & \quad \text{if } h^{\min} \leq h(\mathbf{x}, \mathbf{u}) \text{ is violated.}
 \end{aligned}$$

B.5 Linearization Limit Strategies

The linearization limits are the limits on the control variables in LP. The objective of these limits is to control the linearization error, which increases when the size of the power system increases. For example, if all the linearization limits equal their physical limits, the result of the three-bus system converges but LP solutions for the RTS-79 system and the RTS-96 system are not valid. In addition, linearization limits should be not too small since a small region may require more iterations [78].

The simplest linearization limit strategy is to keep a control variable in a region with a fix ratio to the present value of that variable, but the algorithm using this strategy may need a large number of iterations. Several papers improved this strategy by adjusting

the ratio according to the value of the objective function [79], [80] or the violation of constraints [81]. A proper linearization limit will improve the value of the objective function or reduce the number of violations. However, some control variables may have very small per-unit values, such as real and the reactive powers. If a control variable is close to zero, its linearization limit is very small, so that the variable may be trapped in a small region close to zero. One modified method is using a fixed number, e.g., 0.2, instead of the variable value in calculating its linearization limit. According to the property of the power system, the algorithm can also use the physical limits of a variable instead of using the value as the multiplicand in the calculation since the change of a variable value is not sensitive to its value in the per-unit scale.

In recent years, some advanced linearization limit strategies have been proposed. Schittowsky et al. proposed a strategy based on a decent penalty function [82]. Chen proposed a strategy not including any heuristic criteria [83], [84]. Pourazady and Fu's strategy can reduce the linearization limit exponentially [85]. All these methods are based on the information of the previous iterations. If the penalty function increases, the algorithm raise the linearization limit; otherwise, the algorithm lowers the linearization limit. However, the linearization limit physically depends on the linearization error, which relies on the system configuration and the functions to be linearized. Therefore, Lamberti and Pappalettere proposed several complicated strategies according to the system and function information [86], [88].

The bi-search method considered in this work can compute the linearization limits according to the objective function and all the constraints. At the working point $(\mathbf{x}^o, \mathbf{u}^o, 0)$, the linearized form of the function $f(\mathbf{x}, \mathbf{u}, v)$ is $f_L(\Delta\mathbf{x}, \Delta\mathbf{u}, v)$. η is defined as the maximum allowable linearization error; therefore, $|f(\mathbf{x}, \mathbf{u}, v) - f_L(\Delta\mathbf{x}, \Delta\mathbf{u}, v)| \leq \eta$, which can be represented as a function of the control variable u_i (i.e., $|f(u_i) - f_L(\Delta u_i)| \leq \eta$.) only for simplification. The physical upper bound of u_i is assumed to be $u_i^{max-physical}$, so the

physical upper bound of Δu_i is $\Delta u_i^{\max\text{-physical}} = u_i^{\max\text{-physical}} - u_i^o$. Δu_i^{\min} and Δu_i^{\max} are defined as the lower and the upper bounds of Δu_i respectively.

The bi-search algorithm to find the upper bound Δu_i is shown as follows: (The linearization lower bound can be also obtained similarly.)

1. If $|f(u_i^{\max\text{-physical}}) - f_L(\Delta u_i^{\max\text{-physical}})| \leq \eta$, set $\Delta u_i^{\max\text{-physical}}$ to Δu_i^{\max} and exit.
2. Set the iteration index $i = 0$ and set the maximum iteration index i_{\max} to maintain the precision.
3. Set $u_i^{\text{upper}} = u_i^{\max\text{-physical}}$, $\Delta u_i^{\text{upper}} = \Delta u_i^{\max\text{-physical}}$, $u_i^{\text{lower}} = u_i^o$, and $\Delta u_i^{\text{lower}} = \Delta u_i^o$.
4. Set $u_i^{\text{working}} = 0.5 \cdot u_i^{\text{upper}} + 0.5 \cdot u_i^{\text{lower}}$ and $\Delta u_i^{\text{working}} = 0.5 \cdot \Delta u_i^{\text{upper}} + 0.5 \cdot \Delta u_i^{\text{lower}}$.
5. Compute $f(u_i^{\text{working}})$ and $f_L(\Delta u_i^{\text{working}})$.
6. If $|f(u_i^{\text{working}}) - f_L(\Delta u_i^{\text{working}})| \leq \eta$, keep u_i^{lower} unchanged and set $\Delta u_i^{\text{upper}} = \Delta u_i^{\text{working}}$; otherwise, if $|f(u_i^{\text{working}}) - f_L(\Delta u_i^{\text{working}})| \geq \eta$, set $u_i^{\text{lower}} = u_i^{\text{working}}$ and keep u_i^{upper} unchanged.
7. If $i < i_{\max}$, go to 4; otherwise, set $\Delta u_i^{\text{working}}$ as the linearization upper bound of Δu_i and exit.

Since the bi-search algorithm considers the objective function and all the constraints and therefore requires numerous computations, this method is simplified by considering the power balance equations or the current conservation equations only. The reasons are the following: first, these equations are the key to a valid power flow solution. Second, the updated values after perturbing a control variable are not required since they are constant according to the power flow equations. Therefore, the lower and upper limits on u_i ($u_i \in \mathbf{u}$, $u_i \neq v$) are

$$\Delta u_i^{\min} = \max \left(u_i^{\min} - u_i^o, -\eta / \left(\left. \frac{\partial I_r(\mathbf{x}, \mathbf{u}, v)}{\partial u_i} \right|_{\mathbf{g}_{\text{reduced}}(\mathbf{x}, \mathbf{u}, v)=0} \right), -\eta / \left(\left. \frac{\partial I_i(\mathbf{x}, \mathbf{u}, v)}{\partial u_i} \right|_{\mathbf{g}_{\text{reduced}}(\mathbf{x}, \mathbf{u}, v)=0} \right) \right) \text{ and} \quad (\text{B.8})$$

$$\Delta u_i^{\max} = \min \left(u_i^{\max} - u_i^o, \eta / \left(\left. \frac{\partial I_r(\mathbf{x}, \mathbf{u}, v)}{\partial u_i} \right|_{\mathbf{g}_{\text{reduced}}(\mathbf{x}, \mathbf{u}, v)=0} \right), \eta / \left(\left. \frac{\partial I_i(\mathbf{x}, \mathbf{u}, v)}{\partial u_i} \right|_{\mathbf{g}_{\text{reduced}}(\mathbf{x}, \mathbf{u}, v)=0} \right) \right). \quad (\text{B.9})$$

REFERENCES

- [1] F. Capitanescu and L. Wehenkel, "A new iterative approach to the corrective security-constrained optimal power flow problem," *IEEE Transactions on Power Systems*, Vol. 23, No. 4, Nov. 2008, pp. 1533-1541.
- [2] O. Alsac, J. Bright, M. Prais, and B. Stott, "Further developments in LP-based optimal power flow," *IEEE Transactions on Power Systems*, Vol. 5, No. 3, Aug. 1990, pp. 697-711.
- [3] G. D. Irisarri, X. Wang, J. Tong, and S. Mokhtari, "Maximum loadability of power systems using interior point nonlinear optimization method," *IEEE Transactions on Power Systems*, Vol. 12, No. 1, Aug. 1997, pp. 162-172.
- [4] J. Z. Zhu and M. R. Irving, "Combined active and reactive dispatch with multiple objectives using an analytic hierarchical process," *IEE Proceedings on Generation, Transmission and Distribution*, Vol. 143, Issue: 4, Aug. 1996, pp. 344-352.
- [5] J. A Momoh, R. J. Koessler, M. S. Bond, B. Stott, D. Sun, A. Papalexopoulos, and P. Ristanovic, "Challenges to optimal power flow," *IEEE Transactions on Power Systems*, Vol. 12, No. 1, Feb. 1997, pp. 444-447.
- [6] W. D. Rosehart, C. A. Canizares, and V. H. Quintana, "Multiobjective optimal power flows to evaluate voltage security costs in power networks," *IEEE Transactions on Power Systems*, Vol. 18, No. 2, May 2003, pp. 578-587.
- [7] A. Monticelli, M. V. F. Pereira, and S. Granville, "Security-constrained optimal power flow with post-contingency corrective rescheduling," *IEEE Transactions on Power Systems*, Vol. PWRS-2, No. 1, Feb. 1997, pp. 175-180.
- [8] P. E. Oñate and J. M. Ramirez, "Optimal power flow including transient stability constraints," in *Proceedings of the 2008 IEEE/PES Transmission and Distribution Conference and Exposition*, Apr. 21-24, 2008, pp. 1-9.
- [9] X. Li, Y. Z. Li, and S. H. Zhang, "Analysis of probabilistic optimal power flow taking account of the variation of load power," *IEEE Transactions on Power Systems*, Vol. 23, No. 3, Aug. 2008, pp. 992-999.

- [10] J. Carpentier, "Contribution to the economic dispatch problem," *Bull. Soc. France Elect*, Vol. 8, Aug. 1962, pp. 431-437.
- [11] H. W. Dommel and W. F. Tinney, "Optimal power flow solutions," *IEEE Transactions on Power Apparatus and Systems*, Vol. 87, No. 10, Oct. 1968, pp. 1866-1876.
- [12] B. Stott and E. Hobson, "Power system security control calculations using linear programming," *IEEE Transactions on Power Apparatus and Systems*, Vol. 97, No. 5, Oct. 1978, pp. 1713-1731.
- [13] D. I. Sun, B. Ashley, B. Brewer, A. Hughes, and W. F. Tinney, "Optimal power flow by Newton approach," *IEEE Transactions on Power Apparatus and Systems*, Vol. 103, No. 10, Oct. 1984, pp. 2864-2880.
- [14] A. M. Sasson, "Nonlinear programming solutions for load-flow, minimum-loss, and economic dispatching problems," *IEEE Transactions on Power Apparatus and Systems*, Vol. 88, No. 4, Apr. 1969, pp. 399-409.
- [15] N. S. Rau, "Issues in the path toward an RTO and standard markets," *IEEE Transactions on Power Systems*, Vol. 18, No. 1, May 2003, pp. 435-443.
- [16] R. R. Shoults, S. V. Venkatesh, S. D. Helmick, G. L. Ward, and M. J. Lollar, "A dynamic programming based method for developing dispatch curves when incremental heat rate curves are nonmonotonically increasing," *IEEE Transactions on Power Systems*, Vol. 1, No. 1, Feb. 1986, pp. 10-16.
- [17] T. Gomez, I. J. Pbrez-Aniaga, J. Lumbreras, and V. M. Parra, "A security-constrained decomposition approach to optimal reactive power planning," *IEEE Transactions on Power Systems*, Vol. 6, No. 3, Aug. 1991, pp. 1069-1076.
- [18] R. A. Jabr, "Optimal power flow using an extended conic quadratic formulation," *IEEE Transactions on Power Systems*, Vol. 3, No. 23, Aug. 2008, pp. 1000-1008.
- [19] O. Alsac and B. Stott, "Optimal load flow with steady-state security," *IEEE Transactions on Power Apparatus and Systems*, Vol. 93, No. 3, Jun. 1974, pp. 745-751.
- [20] A. J. Monticelli, M. V. P. Pereira, and S. Granville, "Security-constrained optimal power flow with post-contingency corrective rescheduling," *IEEE Transactions on Power Systems*, Vol. 2, No. 1, Feb. 1987, pp. 175-182.

- [21] J. Martinez-Crespo, J. Usaola, and J. L. Fernandez, "Optimal security-constrained power scheduling by Benders decomposition," *Electric Power Systems Research*, Vol. 77, Issue: 7, May 2007, pp. 739-753.
- [22] T. B. Nguyen and M. A. Pai, "Dynamic security-constrained rescheduling of power systems using trajectory sensitivities," *IEEE Transactions on Power Systems*, Vol. 18, No. 2, May 2003, pp. 848-854.
- [23] C. H. Lin and S. Y. Lin, "Distributed optimal power flow with discrete control variables of large distributed power systems," *IEEE Transactions on Power Systems*, Vol. 23, No. 3, Aug. 2008, pp. 1383-1392.
- [24] K. R. Frisch, *The Logarithmic Potential Method of Convex Programming*, Manuscript at Institute of Economics, University of Oslo, Norway, 1955.
- [25] A. V. Fiacco and G. P. McCormick, *Nonlinear Programming: Sequential Unconstrained Minimization Techniques*, John Willey and Sons, 1968.
- [26] Y. C. Wu, A. S. Debs, and R. E. Marsten, "A direct nonlinear predictor-corrector primal-dual interior point algorithm for optimal power flows," *IEEE Transactions on Power Systems*, Vol. 9, No. 2, May 1994, pp. 876-883.
- [27] S. Granville, "Optimal reactive dispatch through interior point methods," *IEEE Transactions on Power Systems*, Vol. 9, No. 1, Feb. 1994, pp. 136-142.
- [28] G. L. Torres and V. H. Quintana, "An interior-point method for nonlinear optimal power flow using rectangular coordinates," *IEEE Transactions on Power Systems*, Vol. 13, No. 4, Nov. 1998, pp. 1211-1218.
- [29] H. R. Cai, C. Y. Chung, and K. P. Wong, "Application of differential evolution algorithm for transient stability constrained optimal power flow," *IEEE Transactions on Power Systems*, Vol. 23, No. 2, May 1998, pp. 719-728.
- [30] P. E. O. Yumbla, J. M. Ramirez, and C. A. C. Coello, "Optimal power flow subject to security constraints solved with a particle swarm optimizer," *IEEE Transactions on Power Systems*, Vol. 23, No. 1, Feb. 2008, pp. 33-40.
- [31] M. M. El Metwally, A. A. El Emary, F. M. El Bendary, and M. I. Mosaad, "Optimal power flow using evolutionary programming techniques," *12th International Middle-East Power System Conference*, Mar. 12-14 2008, pp. 260-264.

- [32] A. M. Sasson, F. Vitoria, and F. Aboytes, "Optimal load flow solution using the hessian matrix," *IEEE Transactions on Apparatus and Power Systems*, Vol. 92, Issue: 1, Jan. 1973, pp. 31-41.
- [33] R. Billinton and S. S. Sachdeva, "Optimal real and operation in a hydro-thermal system," *IEEE Transactions on Apparatus and Power Systems*, Vol. 91, Issue: 4, July. 1972, pp. 1405-1411.
- [34] R. R. Shoults and D. T. Sun, "Optimal power flow based on P-Q decomposition," *IEEE Transactions on Apparatus and Power Systems*, Vol. 101, Issue: 2, Feb. 1982, pp. 397-405.
- [35] R. A. Ponrajah and F. D. Galiana, "The minimum cost optimal power flow problem solved via the restart homotopy continuation method," *IEEE Transactions on Power Systems*, Vol. 4, No. 1, Feb. 1989, pp. 139-148.
- [36] G. Tognola and R. Bacher, "Unlimited point algorithm for OPF problems," *IEEE Transactions on Power Systems*, Vol. 14, No. 3, Aug. 1999, pp. 1046-1054.
- [37] C. J. Rehn, J. A. Bubenko and D. Sjelvgven, "Voltage optimization using augmented Lagrangian functions and quasi-Newton techniques," *IEEE Transactions on Power Systems*, Vol. 4, No. 4, Dec. 1989, pp. 1470-1483.
- [38] A. M. Sasson, C. Trevino, and F. Aboytes, "Improved Newton's load flow through a minimization technique," *IEEE Transactions on Apparatus and Power Systems*, Vol. 90, Issue: 5, Sept. 1971, pp. 1974-1981.
- [39] M. V. F. Pereira, L. M. V. G. Pinto, S. Granville, and A. Monticelli, "A decomposition approach to security constrained optimal power flow with post contingency corrective rescheduling," *9th Power Systems Computation Conference*, 1987, pp. 585-591.
- [40] C. W. Sanders and C. A. Monroe, "An algorithm for real-time security constrained dispatch," *IEEE Transactions on Power Systems*, Vol. 2, No. 4, Nov. 1987, pp. 175-182.
- [41] G. F. Reid and L. Hasdorff, "Economic dispatch using quadratic programming," *IEEE Transactions on Apparatus and Power Systems*, Vol. 92, Issue: 6, Nov. 1973, pp. 2015-2023.

- [42] M. A. El-Kady, B. D. Bell, V. F. Carvalho, R. C. Burdett, H. H. Happ, and D. R. Vierath, "Assessment of real-time optimal voltage control," *IEEE Transactions and Power Systems*, Vol. 1, Issue: 2, May 1986, pp. 99-107.
- [43] G. F. Reid and L. Hasdorff, "Economic dispatch using quadratic programming," *IEEE Transactions on Apparatus and Power Systems*, Vol. 92, Issue: 6, Nov. 1973, pp. 2015-2023.
- [44] R. C. Burchett, H. H. Happ, and D. R. Vierath, "Quadratically Convergent Optimal Power Flow," *IEEE Transactions on Apparatus and Power Systems*, Vol. 103, Issue: 11, Nov. 1984, pp. 3267-3216.
- [45] F. Capitanescu, M. Glavic, D. Ernst, and L. Wehenkel, "Contingency filtering techniques for preventive security-constrained optimal power flow," *IEEE Transactions on Power Systems*, Vol. 22, No. 4, Nov. 2007, pp. 1690-1697.
- [46] Y. Li and J. D. McCalley, "Decomposed SCOPF for improving efficiency," *IEEE Transactions on Power Systems*, Vol. 24, No. 1, Feb. 2009, pp. 494-495.
- [47] A. M. Geoffrion, "Generalized Benders decomposition," *Journal of Optimization Theory Applications*, Vol. 10, No. 4, Oct. 1972, pp. 237-260.
- [48] M. Shahidehopour and Y. Fu, "Benders decomposition: applying Benders decomposition to power systems," *IEEE Power and Energy Magazine*, Vol. 3, No. 1, Mar. 2005, pp. 20-21.
- [49] J. Martinez-Crespo, J. Usaola, and J. L. Fernandez, "Security-constrained optimal generation scheduling in large-scale power systems," *IEEE Transactions on Power Systems*, Vol. 21, No. 1, Feb. 2006, pp. 494-495.
- [50] G. Hug-Glanzmann and G. Andersson, "Decentralized optimal power flow control for overlapping areas in power systems," *IEEE Transactions on Power Systems*, Vol. 24, No. 1, Feb. 2009, pp. 327-336.
- [51] D. W. Wells, "Method for economic secure loading of a power system," in *Proceedings of the Institution of Electrical Engineers*, Vol. 115, No. 8, Aug. 1968, pp. 1190-1194.
- [52] C. M. Shen and M. A. Laughton, "Power system load scheduling with security constraints using dual linear programming," in *Proceedings of the Institution of Electrical Engineers*, Vol. 117, No. 11, Nov. 1970, pp. 2117-2127.

- [53] B. Stott and J. L. Marinho, "Linear programming for power system network security applications," *IEEE Transactions on Apparatus and Power Systems*, Vol. 98, Issue: 3, May 1979, pp. 837-848.
- [54] M. Santos-Neito and V. H. Quintana, "Linear Reactive Power Studies for Longitudinal Power Systems," in *Proceedings of 9th Power Systems Computation Conference*, 1987, pp. 783-787.
- [55] R. Mota-Palomino and V. H. Quintana, "Sparse reactive power scheduling by a penalty-function - linear programming technique," *IEEE Transactions on Power Systems*, Vol. 1, No. 3, Aug. 1986, pp. 31-39.
- [56] S. A. Farghal, M. A. Tantawy, M. S. Abou-Hussein, S. A. Hassan, and A. A. Abou-Slela, "A fast technique for power system security assessment using sensitivity parameters of linear programming," *IEEE Transactions on Apparatus and Power Systems*, Vol. 103, Issue: 5, May 1984, pp. 946-953.
- [57] K. A. Clements, P. W. Davis, K. D. Frey, S. A. Hassan, and A. A. Abou-Slela, "An interior point algorithm for weighted least absolute value power system state estimation," in *Proceedings of IEEE/PES 1991 Winter Meeting*.
- [58] S. Granville and F. R. de M. Alves, "Active-reactive coupling in optimal reactive dispatch, a solution via Karush-Kuhn-Tucker optimality conditions," *IEEE Transactions on Power Systems*, Vol. 9, No. 4, Nov. 1994, pp. 1774-1779.
- [59] L. S. Vargas, V. H. Quintana, A. Vannelli, "A tutorial description of an interior point method and its application to security-constrained economic dispatch," *IEEE Transactions on Power Systems*, Vol. 8, No. 3, Aug. 1993, pp. 1315-1324.
- [60] R.A Jabr, A.H. Coonick, and B.J. Cory, "A primal-dual interior point method for optimal power flow dispatching," *IEEE Transactions on Power Systems*, Vol. 17, No. 3, Aug. 2002, pp. 654-662.
- [61] Y. C. Wu, A. S. Debs, R. E. Marsten, "A direct nonlinear predictor-corrector primal-dual interior point algorithm for optimal power flows," *IEEE Transactions on Power Systems*, Vol. 9, No. 2, May 1994, pp. 876-883.
- [62] G. L. Torres and V. H. Quintana, "On a nonlinear multiple-centrality-corrections interior-point method for optimal power flow," *IEEE Transactions on Power Systems*, Vol. 16, No. 2, May 2001, pp. 222-228.

- [63] S. Dan, "Multi-area economic dispatch with tie line constraints," *IEEE Transactions on Power Systems*, Vol. 10, No. 4, Nov. 1995, pp. 1946-1951.
- [64] A. G. bakirtzis, P. N. Biskas, C. E. Zoumas, and V. Petridis, "Optimal power flow by enhanced genetic algorithm," *IEEE Transactions on Power Systems*, Vol. 17, No. 2, May 2002, pp. 229-236.
- [65] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of IEEE International Conference on Neural Networks*, Vol. 4, Nov. 27 - Dec. 1 1995, pp. 1942-1948.
- [66] M. A. Abido, "Optimal power flow using particle swarm optimization," in *Proceedings of the International Journal of Electrical Power and Energy Systems*, Vol. 24, No. 7, Oct. 2002, pp. 563-571.
- [67] B. Zhao, C. X. Guo, and Y. J. Cao, "Improved particle swarm optimization algorithm for OPF problem," in *Proceedings of the IEEE PES Power Systems Conference and Exposition*, Vol. 1, Oct. 10-13 2004, pp. 233-238.
- [68] S. He, J. Y. Wen, E. Prempain, Q. H. Wu, J. Fitch, and S. Mann, "An improved particle swarm optimization for optimal power flow," *International Conference on Power System Technology*, Vol. 2, Nov. 21-24 2004, pp. 1633-1637.
- [69] J. G. Vlachogiannis and K. Y. Lee, "A comparative study on particle swarm optimization for optimal steady-state performance of power systems," *IEEE Transactions on Power Systems*, Vol. 21, No. 4, Nov. 2006, pp. 1718-1728.
- [70] J. G. Vlachogiannis and K. Y. Lee, "Coordinated aggregation particle swarm optimization applied in reactive power and voltage control," in *Proceedings of the IEEE Power Engineering Society General Meeting*, Jun. 18-22 2006.
- [71] A. Berizzi, C. Bovo, M. Merlo, G. Callegari, M. Porcellini, and M. Pozzi, "Second order sensitivities for constrained reactive optimal power flow," in *Proceedings of the 43rd International Universities Power Engineering Conference*, Sept. 1-4 2006, pp. 1-7.
- [72] G. L. Torres and V. H. Quintana, "Rectangular form optimal power flow by interior-point methods," in *Proceedings of COPIMERA '97*, Sept. 28 - Oct. 3 1997, pp. 64-70.

- [73] A. P. S. Meliopoulos, *Power System Modeling, Analysis, and Control*, Georgia Institute of Technology, 2008.
- [74] G. K. Stefopoulos and A. P. S. Meliopoulos, "Quadratized three-phase induction motor model for steady-state and dynamic analysis," in *Proceedings of 38th North American Power Symposium*, Sept. 17-19 2006, pp. 65-75.
- [75] F. Yang, A. P. S. Meliopoulos, G. J. Cokkinides, and G. K. Stefopoulos, "Contingency simulation using single phase quadratized power flow," in *Proceedings of International Conference on Probabilistic Methods Applied to Power Systems*, Jun. 11-15 2006, pp. 1-8.
- [76] P.M. Subcommittee, "IEEE reliability test system," *IEEE Transactions on Apparatus and Power Systems*, Vol. 98, Issue: 6, Nov. 1979, pp. 2047-2054.
- [77] P.M. Subcommittee, "IEEE reliability test system - 1996," *IEEE Transactions on Power Systems*, Vol. 14, No. 3, Aug. 1999, pp. 1010-1020.
- [78] G. G. Pope, "Optimum design of stressed skin structures," *AIAA Journal*, Vol. 11, 1973, pp. 1545-1552.
- [79] R. T. Haftka and Z. Gurdal, *Elements of Structural Optimization*, 3rd edition, Dordrecht: Kluwer Academic Publishers, 1992.
- [80] B. A Wujek and J. E Renaud, "New adaptive move-limit management strategy for approximate optimization," Part 1 and 2, *AIAA Journal*, Vol. 36, 1998, pp. 1911-1934.
- [81] G. N Vanderplaats and S. Kodyalam, "Two level approximation methods for stress constraints in structural optimization," *AIAA Journal*, Vol. 28, 1990, pp. 948-951.
- [82] K. Schittowski, C. Zillober, and R. Zotemantel, "Numerical comparison of nonlinear programming algorithms for structural optimization," *Structural and Multidisciplinary Optimization*, Vol. 7, No. 1-2, Feb. 1994, pp. 1-19.
- [83] T. Y. Chen, "Calculation of the move limits for the sequential linear programming method," *International Journal for Numerical Methods in Engineering*, Vol. 36, No. 15, Jun. 2005, pp. 2661-2619.
- [84] T. Y. Chen, "A comprehensive solution for enhancing the efficiency and the robustness of the SLP algorithm," *Structural and Multidisciplinary Optimization*, Vol. 66, No. 4, Feb. 1998, pp. 373-384.

- [85] M. Pourazady and Z. Fu, "An integrated approach to structural shape optimization," *Structural and Multidisciplinary Optimization*, Vol. 60, No. 2, July 1996, pp. 279-289.
- [86] L. Lamberti and C. Pappalettere, "Comparison of the numerical efficiency of different sequential linear programming based algorithms for structural optimisation problems," *Computers and Structures*, Vol. 7, No. 6, July 2000, pp. 713-728.
- [87] L. Lamberti and C. Pappalettere, "Move limits definition in structural optimization with sequential linear programming," Part I and II, *Computers and Structures*, Vol. 81, No. 4, Mar. 2003, pp. 197-238.
- [88] L. Lamberti and C. Pappalettere, "Improved sequential linear programming formulation for structural weight minimization," *Computer Methods in Applied Mechanics and Engineering*, Vol. 193, Issues: 33-35, Aug. 2004, pp. 3493-3521.
- [89] G. F. Reid and L. Hasdorff, "Economic dispatch using quadratic programming," *IEEE Transactions on Apparatus and Power Systems*, Vol. 92, Issue: 6, Nov. 1973, pp. 2015-2023.
- [90] Q. B. Dam, A. P. Meliopoulos, G. T. Heydt, and A. Bose, "A breaker-oriented, three-phase IEEE 24-substation test system," *IEEE Transactions on Power Systems*, Vol. 25, Issue: 1, Jan. 2010, pp. 59-67.
- [91] R. Huang, F. Evangelos, G. J. Cokkinides, and G. K. Stefopoulos, "Substation based dynamic state estimator - numerical experiment," in *Proceedings of 2010 IEEE PES Transmission and Distribution Conference and Exposition*, Apr. 19-22 2010, pp. 1-8.
- [92] F. Evangelos, R. Huang, G. J. Cokkinides, and G. K. Stefopoulos, "Implementation of a 3-phase state estimation tool suitable for advanced distribution management systems," in *Proceedings of 2010 IEEE PES Power Systems Conference and Exposition*, Mar. 20-23 2011, pp. 1-8.
- [93] J. L. Hennessy and D. A. Patterson, *Computer Architecture: A Quantitative Approach*, 4th ed., Morgan Kaufmann, San Francisco, CA, Sept. 27 2007, pp. 216.

- [94] Y. Hong and F. Wang, "Development of three-phase Newton optimal power flow for studying imbalance/security in transmission systems," *Electric Power Systems Research*, Vol. 55, Issue: 1, Jul. 2000, pp. 39-48.
- [95] H. M. Khodr, M. A. Matos, and J. Pereira, "Distribution optimal power flow," in *Proceedings of 2007 IEEE Power Tech*, Jul. 1-5 2007, pp. 1-5.
- [96] Y. Zhu and K. Tomsovic, "Optimal distribution power flow for systems with distributed energy resources," *Electrical Power and Energy Systems*, Vol. 29, Issue: 3, Nov. 2007, pp. 260-267.
- [97] G. P. Harrison, A. Piccolo, P. Siano, and A. R. Wallace, "Hybrid GA and OPF evaluation of network capacity for distributed generation connections," *Electric Power Systems Research*, Vol. 78, Issue: 3, Nov. 2008, pp. 392-398.
- [98] M. J. Dolan, E. M. Davidson, G.W. Ault, F. Coffele; I. Kockar, and J.R. McDonald, "Using optimal power flow for management of power flows in active distribution networks within thermal constraints," in *Proceedings of 44th International Universities Power*, Sept. 1-4 2009, pp. 1-5.
- [99] A. R. Ahmadi and T. C. Green, "Optimal power flow for autonomous regional active network management system," in *Proceedings of 2009 IEEE PES General Meeting*, Jul. 26-30 2009, pp. 1-7.
- [100] L. F. Ochoa, C. J. Dent, and G. P. Harrison, "Distribution network capacity assessment: variable DG and active networks," *IEEE Transactions on Power Systems*, Vol. 25, Issue: 1, Nov. 2010, pp. 87-95.
- [101] L. F. Ochoa and G. P. Harrison, "Minimizing energy losses: optimal accommodation and smart operation of renewable distributed generation," *IEEE Transactions on Power Systems*, Vol. 26, Issue: 1, Nov. 2011, pp. 198-205.
- [102] S. Bruno, S. Lamonaca, G. Rotondo, U. Stecchi, and M. La Scala, "Unbalanced three-phase optimal power flow for smart grids," *IEEE Transactions on Industrial Electronics*, Vol. PP, Issue: 99, Nov. 2011, pp. 1-10.
- [103] *OpenDSS*, available at <http://electricdss.sourceforge.net/>.
- [104] A. Merlin and H. Back, "Search for a minimal-loss operating spanning tree configuration in an urban power distribution system," in *Proceedings of the 5th Power System Computation Conference (PSCC)*, Cambridge, 1975, pp. 1-18.

- [105] D. Shirmohammadi and H. W. Hong, "Reconfiguration for electric distribution networks for resistive line loss reduction," *IEEE Transactions on Power Delivery*, Vol. 4, Issue: 2, Apr. 1989, pp. 1492-1498.
- [106] N. G. Caicedo, C. A. Lozano, J. F. Díaz, C. Rueda, G. Gutiérrez, C. Olarte, "Loss reduction in distribution networks using concurrent constraint programming," in *Proceedings of 2004 International Conference on Probabilistic Methods Applied to Power Systems*, Sept. 12-16 2004, pp. 295-300.
- [107] M. A. Matos and P. Melo, "Multiobjective reconfiguration for loss reduction and service restoration using simulated annealing," in *Proceedings of 1999 International Conference on Power Tech*, Aug. 29 - Sept. 2 1999, pp. 213.
- [108] M. A. Matos and P. Melo, "Loss minimization in distribution networks with multiple load scenarios," in *Proceedings of 2001 International Conference on Power Tech*, Sept. 10- 13 2001, pp. 5-11.
- [109] M. A. Matos et al., "Meta-heuristics Applied to Power Systems," *Meta heuristics: Computer Decision-Making*, Kluwer Academic Publishers B. V., Nov. 30 2003, pp. 449-464.
- [110] A. J. Wood and B F. Wollenberg, *Power Generation, Operation, and Control*, John Wiley & Sons, Jan. 1996, pp. 104.
- [111] J. H. Chow, D. K. Frederick, and N. W. Chbat, *Discrete-Time Control Problems Using MATLAB*, CL-Engineering, Oct. 7 2002, pp. 70.
- [112] *IEEE test cases*, available at <http://www.ee.washington.edu/research/pstca/>.
- [113] O. Alsac and B. Stott, "Optimal load flow with steady state security," *IEEE Transactions on Apparatus and Power Systems*, Vol. 93, Issue: 3, Nov. 1974, pp. 745-751.
- [114] G. W. Bills, et al., "On-line stability analysis study," *RP90-1 Report for the Edison Electric Institute*, Oct. 12 1974, pp. 1-20 - 1-35.
- [115] M. A. Pai, *Energy Function Analysis for Power System Stability*, Kluwer Academic Publishers, Boston, Aug. 31 1989.
- [116] T. Athay, R. Podmore, and S. Virmani, "A Practical Method for the Direct Analysis of Transient Stability," *IEEE Transactions on Apparatus and Power Systems*, Vol. 98, Issue: 2, Nov. 1979, pp. 573-584.

- [117] *MATPOWER*, available at <http://www.pserc.cornell.edu/matpower/>.
- [118] *TSPOPF*, available at <http://www.pserc.cornell.edu/tspopf/>.
- [119] *MINOPF*, available at <http://www.pserc.cornell.edu/minopf/>.
- [120] H. Wang, C. E. Murillo-Sánchez, R. D. Zimmerman, and R. J. Thomas, "On computational issues of market-based optimal power flow," *IEEE Transactions on Power Systems*, Vol. 22, No. 3, Aug. 2007, pp. 1185-1193.
- [121] H. Wang, *On the Computation and Application of Multi-period Security-constrained Optimal Power Flow for Real-time Electricity Market Operations*, Ph.D. thesis, Electrical and Computer Engineering, Cornell University, May 2007.
- [122] *GLPK*, available at <http://www.gnu.org/s/glpk/>.
- [123] *Gurobi*, available at <http://www.gurobi.com/>.
- [124] K. G. Murty, "A new practically efficient interior point method for convex quadratic programming," *Mathematical Programming and Game Theory for Decision Making*, World Scientific Publishing, Hackensack, NJ, Apr. 2008, pp. 21-31.
- [125] F. Delbos and J. C. Gilbert, "Global linear convergence of an augmented Lagrangian algorithm for solving convex quadratic optimization problems," *Journal of Convex Analysis*, Vol. 12, No. 1, 2005, pp. 45-69.
- [126] D. P. O'Leary, "A generalized conjugate gradient algorithm for solving a class of quadratic programming problems," *Linear Algebra and its Applications*, Vol. 34, Dec. 1980, pp. 371-399.
- [127] K. G. Murty, *Linear Complementarity, Linear and Nonlinear Programming*, Sigma Series in Applied Mathematics, Berlin, 1988.
- [128] I. Bernhardt, N.M. Fraser, E.M. Jewkes, and M. Tajima, *Engineering Economics in Canada*, 3rd ed., Pearson Prentice Hill, Toronto, 2006.
- [129] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, 2nd ed., The MIT Press, Cambridge, 2001.
- [130] R. Bellman, *Dynamic Programming*, Princeton University Press, Princeton, NJ, 1957a.

- [131] K. D. W. Nandalal and J. J. Bogardi, *Dynamic Programming Based Operation of Reservoirs: Applicability and Limits*, Cambridge University Press, Cambridge, UK, 2007.

VITA

YE TAO

YE TAO was born in Xi'an, China in 1980. He attended public schools in Xi'an and received his high-school diploma in June 1999. In September 1999, he joined Tsinghua University in Beijing, China, where he received a B.E. in Automatic Control in 2003 and a M.E. in System Engineering in 2006. He then moved to the United States and joined the Georgia Institute of Technology, in Atlanta, Georgia, in Fall 2006, receiving a M.S. in Electrical and Computer Engineering in 2008, before continuing to pursue a doctorate in Electrical Engineering, again at Georgia Tech.